



## Database

**Alunni:** Classe V A Sistemi Informativi Aziendali, indirizzo Tecnico Economico "A. Guarasci" Rogliano, dell'Istituto Istruzione Superiore IPSIA "Marconi" Cosenza –Lic Sc.e ITE Rogliano

- Altomare Francesco
- (in collaborazione con la classe V A Sistemi informatici Aziendali, indirizzo Tecnico Economico "A. Guarasci" Rogliano dell'Istituto Superiore IPSIA "Marconi" Cosenza – Lic. Sc. e ITE Rogliano)
- Bruni Mihaela
- Fuoco Jessica
- Gervasi Christian
- Reda Maria Vittoria
- Rizzo Simone
- Rukovska Zoryana
- Sottile Antonio
- Spinelli Francesca
- Vena Valentina
- Viscomi Nicola
- Vizza Andrea
- Vizza Carmen
- Vizza Genny
- Vizza Martina

**Docente referente:** Prof.ssa Rosa Marincola



## INTRODUZIONE

Nelle scuole, nelle università, nelle banche, negli uffici, nelle aziende, nelle biblioteche, quando occorre gestire un insieme organizzato di dati di grandi dimensioni per il supporto allo svolgimento di attività in modo automatico, vengono utilizzati i database (o banche dati). In questo lavoro presenteremo le loro caratteristiche salienti, le fasi della loro progettazione e implementazione, evidenziando l'importanza della matematica in questo settore. Presenteremo i modelli più diffusi di banche dati e ci soffermeremo su modello relazionale che si basa sulla teoria degli insiemi e sulla logica del primo ordine. Tale modello è strutturato intorno al concetto matematico di relazione (detta anche tabella), per il suo trattamento ci si avvale di strumenti quali il calcolo relazionale e l'algebra relazionale. Anche altri modelli si basano sulla teoria dei grafi e degli alberi, dunque è la matematica che fornisce la teoria e gli strumenti per la creazione dei modelli dei sistemi informatici che ormai pervadono la nostra quotidianità.

### **Sistemi informativi ed informatici**

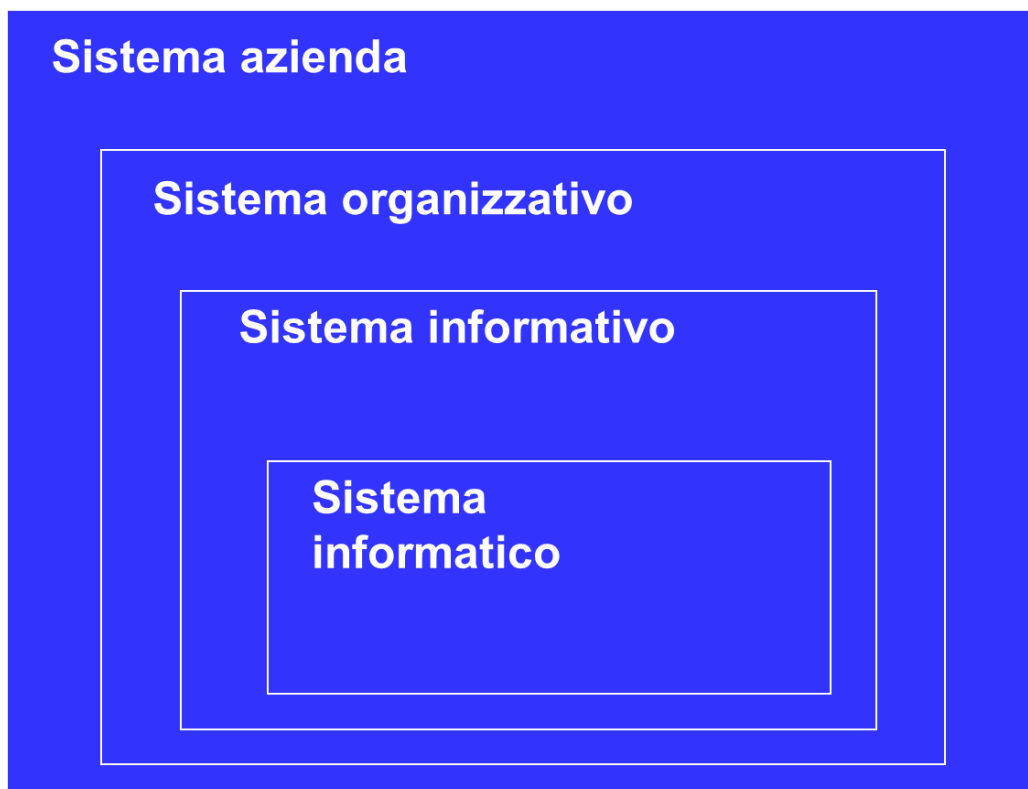
Ogni impresa o istituzione, pubblica o privata, si struttura e si organizza secondo la propria missione e per perseguire gli obiettivi generali o target. La natura degli obiettivi varia a seconda della tipologia di impresa. Ad esempio per

un'azienda industriale l'obiettivo principale è quello di ottenere degli utili attraverso la progettazione, produzione e vendita di beni; per un'azienda di servizi di produzione, l'obiettivo è quello di fornire alle migliori condizioni di qualità e di costo un insieme di servizi. Mission o obiettivi sono diversi, infatti per un'azienda di produzione scarpe la mission è "produrre scarpe", mentre gli obiettivi possono essere quello di ottenere un certo fatturato o l'ingresso in certi mercati. Oltre alle nozioni tradizionali di struttura di un'azienda, ossia di suddivisione dell'organizzazione in specifiche unità organizzative, alle quali sono attribuiti determinati compiti e specifici obiettivi hanno un ruolo essenziale le risorse e i processi. Una risorsa è tutto ciò con cui l'azienda opera per perseguire i suoi obiettivi generali. Le risorse di una azienda (o ente, amministrazione) sono di diversa natura:

- persone
- denaro
- materiali
- informazioni.

Il processo è un insieme di attività che l'organizzazione deve svolgere per gestire le sue risorse.

Ogni azienda può essere considerata come un sistema che contiene altri sottosistemi, in modo schematico essi si possono rappresentare mediante insiemi, come nella seguente figura.



Un sistema organizzativo è un insieme di risorse e regole per lo svolgimento coordinato delle attività (processi) al fine del perseguimento degli scopi. Un

sistema informativo è un componente (sottosistema) di una organizzazione che gestisce (acquisisce, elabora, conserva, produce) le informazioni di interesse (cioè utilizzate per il perseguimento degli scopi dell'organizzazione). Il concetto di "sistema informativo" è indipendente da qualsiasi automatizzazione, esistono infatti organizzazioni (per esempio servizi anagrafici e banche) e che operano da secoli. Il sistema informatico è la porzione automatizzata del sistema informativo, cioè è la parte del sistema informativo che gestisce informazioni con tecnologia informatica. Nei sistemi informatici (e non solo), le informazioni vengono rappresentate in modo essenziale attraverso i dati. Un dato è quanto è immediatamente presente alla conoscenza, prima di ogni elaborazione. Un'informazione è l'incremento di conoscenza che un dato può dare. I dati per avere significato, devono essere interpretati; la mente umana tratta le informazioni, i computer e le macchine invece elaborano solo dati. Ad esempio, se scriviamo il numero "8" su un foglio, esso è un dato grezzo, da solo non significa molto, ma se sul foglio scriviamo anche una domanda, ad esempio "Quanti anni ha Matteo?", quel numero acquista significato. Allora i dati possono essere interpretati per fornire informazione e arricchire la conoscenza.

## LA TEORIA DELLE BASI DI DATI

La teoria delle basi di dati studia come organizzare al meglio grandi quantità di dati per poterli gestire in modo semplice, efficiente, efficace, sicuro.

Una base di dati (o database) è una raccolta di dati strutturati, progettata per essere accessibile da parte di applicazioni e utenti differenti e deve essere:

- sicura: progettata in modo che non subisca danni a causa accidentali o interventi non autorizzati;
- integra: deve garantire che le operazioni eseguite dagli utenti autorizzati non provochino perdita di consistenza di dati;
- consistente: i dati devono essere significativi e utilizzabili nelle varie applicazioni utilizzate dall'azienda;
- condivisibile: diversi utenti e applicazioni devono poter accedere ai dati comuni;
- persistente: i dati devono essere salvati su memoria di massa affinché durino nel tempo;
- scalabile: deve mantenere intatte le proprie performance all'aumentare della quantità dei dati.

Nello sviluppo della teoria dei database, dal 1960 in poi, sono emersi principalmente diversi tipi di modelli di database, basati su **modelli** matematici.

### Modello Gerarchico

Particolarmente adatto per rappresentare situazioni nelle quali è possibile fornire all'insieme dei dati una struttura nella quale ci sono entità collegate secon-

do uno schema ad albero, nel quale i nodi rappresentano le entità e gli archi rappresentano le associazioni. Nella pratica l'entità è un file, l'istanza è un record e gli attributi sono i campi del record. Presenta dei limiti soprattutto nella rigidità della struttura di dati creata, che talvolta non riesce ad evitare ridondanze dei dati (dati ripetuti).

### **Modello Reticolare**

In questo modello le entità rappresentano i nodi e le associazioni rappresentano gli archi di uno schema a grafo orientato: si tratta cioè di una estensione del modello gerarchico. Non esiste quindi una gerarchia predefinita tra le entità, un record figlio può avere un numero qualsiasi di padri: in questo modo vengono evitate situazioni di ripetizione di dati uguali.

Risulta però più difficile l'implementazione e la costruzione del software applicativo. Ha avuto una discreta diffusione negli anni '70 soprattutto nei sistemi di grandi dimensioni.

### **Modello Relazionale**

Rappresenta il database come un insieme di tabelle. Esso viene considerato attualmente il modello più semplice ed efficace, perché è più vicino al modo consueto di pensare i dati e si adatta in modo naturale alla classificazione e alla strutturazione dei dati.

Il modello relazionale nasce nel 1970, proposto da Edward F. Codd, ricercatore dell'IBM come idea di un modello logico molto semplice e nello stesso tempo in grado di superare i limiti degli altri modelli utilizzati. Entra a far parte di sistemi commerciali a partire dal 1978. Esso si basa su alcuni concetti fondamentali tipicamente matematici e assegna grande importanza all'uso rigoroso del linguaggio matematico, con due obiettivi importanti:

- utilizzare un linguaggio conosciuto a livello universale, qual è il linguaggio matematico,
- eliminare i problemi di ambiguità nella terminologia e nella simbologia.

### **Modello a oggetti**

Rappresenta il modello attualmente utilizzato nelle applicazioni multimediali.

## **LA PROGETTAZIONE DI UNA BASE DI DATI**

La metodologia di progettazione di una base di dati è un insieme di attività tra loro collegate al fine di realizzare un'astrazione di un problema reale. Per la costruzione di un database le fasi di progettazione distinte sono tre: la **progettazione concettuale**, il cui scopo è quello di costruire e definire una corretta e completa realtà d'interesse. La **progettazione logica** ha lo scopo di trasformare la rappresentazione astratta dello schema logico creato nella

progettazione concettuale in uno schema logico riassumibile con relazioni create tra le tabelle. La **progettazione fisica** ha come scopo di trasportare lo schema logico in un supporto di memoria di massa.

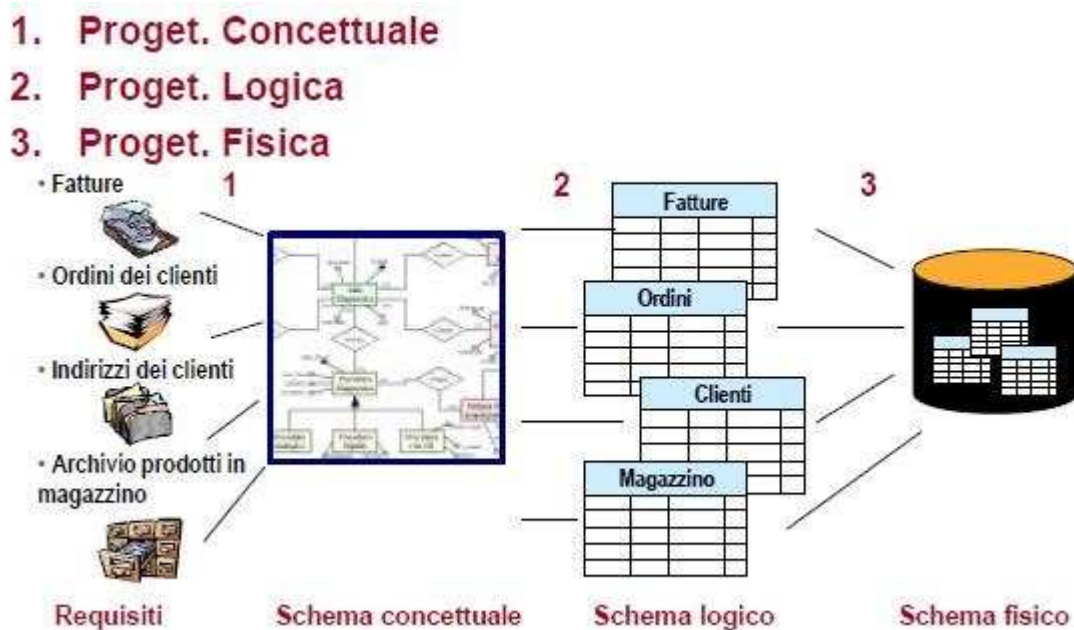


Figura 1 <http://www.media24ore.com/sites/default/files/products/progettazioneclf1.jpg>

La **progettazione concettuale** consiste nel riorganizzare tutti gli elementi presenti nella documentazione delle specifiche del problema iniziale per rappresentare la realtà di interesse in termini di una descrizione formale e completa. Si tratta di un modello grafico per la descrizione dei dati e delle loro relazioni all'interno di una realtà di interesse. Il risultato di questo lavoro è rappresentato dal diagramma ER (Entity Relationship). Per rappresentare un diagramma ER si parte dal fatto che la realtà da rappresentare è composta da **entità** caratterizzate da specifici attributi. Le entità sono ciò che esiste all'interno della realtà che si vuole modellare, sono classi di oggetti, nel diagramma ER sono sostantivi che si rappresentano in dei rettangoli.

Gli **attributi** descrivono le caratteristiche delle specifiche entità. Il procedimento mentale che permette di evidenziare alcune proprietà prende il nome di processo di astrazione. Un tipo di entità descrive la struttura di tutti gli elementi di un certo insieme. Ogni singolo esemplare che appartiene a un tipo di entità è detto istanza.

Le proprietà delle entità sono descritte tramite gli attributi. Le proprietà elementari vengono descritte attraverso gli attributi semplici. Ogni attributo è specificato da:

- Un nome
- Un formato, che indica il tipo di valori che può assumere
- Una dimensione, che indica la quantità massima di caratteri o cifre inseribili

- Un valore, i diversi valori assunti dagli attributi indicano le diverse istanze delle entità. L'insieme dei possibili valori si chiama dominio dell'attributo.
- Una opzionalità, che indica la possibilità di non essere sempre valorizzato.

Gli attributi composti sono costruiti mediante aggregazioni di attributi eterogenei che a loro volta possono essere semplici o frutto di altre aggregazioni. Gli attributi multipli sono inseriti in un elenco di lunghezza variabile di attributi dello stesso tipo. Gli attributi nel diagramma ER si rappresentano collegati mediante linee alle entità (o alle associazioni cui appartengono).

Le **associazioni** (dette anche relazioni) rappresentano un legame tra due o più entità. Il numero di entità legate è indicato dal grado dell'associazione: un buono schema ER è caratterizzato da una prevalenza di associazioni con grado due.

Altre caratteristiche sono:

- associazione diretta: associazione A dall'entità X a quella Y
- inversa: associazione  $A^{-1}$  da Y a X
- partecipazione: può essere totale o parziale a seconda dell'opzionalità del legame tra le entità
- molteplicità: indica il numero massimo di istanze dell'entità X che possono trovarsi in relazione con un'istanza dell'entità Y, e viceversa. Si parla di associazione univoca da X a Y se ogni istanza di X può trovarsi in relazione con al massimo un'istanza di Y. Se tale vincolo non esiste, si dice che l'associazione è multivalore o multipla.
- cardinalità: descrive parallelamente la molteplicità dell'associazione diretta e della sua inversa. Si hanno tre casi:
  - 1:1 detta biunivoca
  - 1:N (si legge 1 a molti) oppure N:1 detta semplice
  - N:N (si legge molti a molti) detta complessa

Vengono rappresentate graficamente da un rombo. Il nome può essere un verbo in modo da fornire una direzione di lettura.

Se ad esempio si vuole creare un database per gestire un campionato di calcio diviso in serie ed eventualmente in gironi, in modo che siano accessibili il calendario di tutte le partite, i risultati e le classifiche per ogni serie e girone.

Dopo aver analizzato il problema, si può costruire il diagramma ER in figura 2.

Si definisce vincolo di integrità una proprietà che deve essere soddisfatta dalle istanze al fine di mantenere la correttezza delle informazioni (per una classificazione dettagliata si veda

[http://it.wikipedia.org/wiki/Vincolo\\_di\\_integrit%C3%A0](http://it.wikipedia.org/wiki/Vincolo_di_integrit%C3%A0)). Ad esempio un vincolo nel precedente problema è che il numero dei goal segnati non può essere negativo, in simboli si scrive:

V1. Calciatore.GoalSegnati $\geq$ 0.

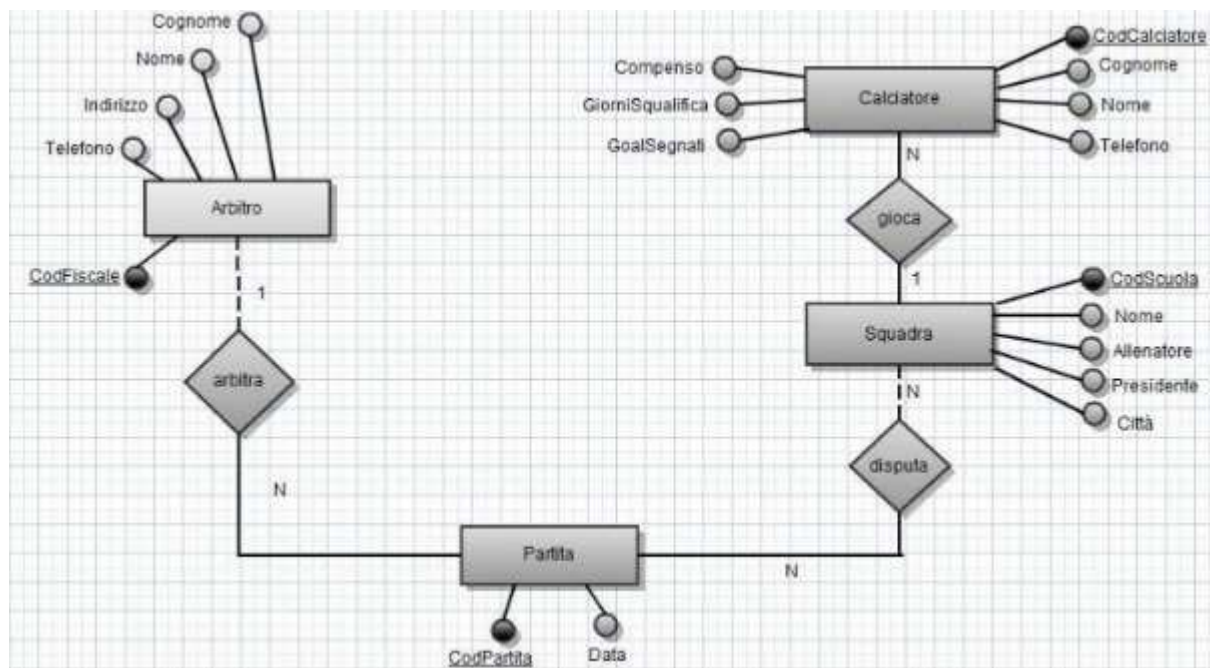


Fig. 2 <http://www.mercurio5a.altervista.org/joomla/images/diagramma%20er%20angelo.jpg>

Dal diagramma ER, mediante una serie di regole di derivazione (si veda [http://scuola.linux.it/docs/fb\\_db/database.html#livello\\_logico](http://scuola.linux.it/docs/fb_db/database.html#livello_logico)), si ottiene lo schema logico-relazionale (fase di **progettazione logica**).

Richiamiamo la definizione di **relazione matematica**: siano  $D_1, \dots, D_n$  ( $n$  insiemi anche non distinti detti domini), dato il prodotto cartesiano  $D_1 \times \dots \times D_n$  una relazione  $r \subseteq D_1 \times D_2$  è un insieme di  $n$ -uple ordinate  $(d_1, \dots, d_n)$  tali che  $d_1 \in D_1, \dots, d_n \in D_n$

A ogni dominio è associato un nome che lo distingue dagli altri all'interno della relazione; ed  $n$  è detto grado che si indica con  $\text{Grado}(r)$ . Gli elementi di una relazione sono detti  $t$ -uple o  $n$ -ple. Una relazione è caratterizzata dall'istanza che è l'insieme delle  $t$ -uple in un determinato istante di tempo, il numero di  $t$ -uple presenti in un dato istante viene detto cardinalità e viene indicato con  $\text{Card}(R)$ .

Una relazione può essere rappresentata anche nei seguenti modi:

- per elencazione
- in forma tabellare
- in forma insiemistica (con i diagrammi di Eulero-Venn)

Per ciascuna relazione si deve specificare la presenza di una chiave, esattamente si dice chiave primaria candidata un insieme non vuoto di attributi attraverso i quali è possibile individuare in maniera univoca una  $t$ -upla per ogni istanza della relazione. In genere una relazione può ammettere diverse chiavi



candidate, ma fra queste ne viene scelta solo una col minor numero di attributi chiamata chiave primaria.

Per il campionato di calcio lo schema logico-relazionale è:

CALCIATORE (CodCalciatore, Cognome, Nome, Telefono, Compenso, GiorniSqualificati, GoalSegnati, CodSquadra)

SQUADRA (CodSquadra, Nome, Allenatore, Presidente, Città)

ARBITRO (CodFiscale, cognome, Nome, Via, NumCivico, CAP, Luogo, Provincia Telefono)

PARTITA (CodPartita, Data, CodFiscale)

DISPUTA (CodSquadra, CodPartita)

Durante lo studio del modello relazionale, nella fase di progettazione logica è stato necessario richiamare nozioni di insiemistica. Questo ci ha consentito di integrare nel nostro lavoro il nostro compagno diversabile che, cooperando in gruppo con altri membri della classe, ha fornito il suo contributo attraverso rappresentazioni per elencazione e con i diagrammi di Eulero-Venn. Altre foto oltre a quelle seguenti sono reperibili al link:

<https://sites.google.com/site/francescoaltomare2/home/una-nuova-era/unnuovolavorosullinsiemistica>



## LE OPERAZIONI RELAZIONALI

Le operazioni relazionali sono in totale 7:

- unione
- intersezione
- differenza
- proiezione
- restrizione
- prodotto cartesiano
- giunzione naturale

### Unione

Si effettua tra due relazioni R e S compatibili (stesso numero di attributi e stesso tipo di attributi nelle stesse posizioni).

Restituisce semplicemente l'unione insiemistica delle due relazioni:

$$R \cup S = \{t | t \in R \text{ OR } t \in S\}$$

### Intersezione

Si effettua tra due relazioni R e S compatibili.

Restituisce le t-uple in comune sia ad R che ad S:

$$R \cap S = \{t | t \in R \text{ AND } t \in S\}$$

### Differenza

Si effettua tra due relazioni R e S compatibili.

Restituisce le t-uple di R che non sono presenti anche in S.

$$R - S = \{t | t \in R \text{ AND } t \notin S\}$$

### Proiezione

Data una relazione R ed un sottoinsieme

$$A = \{A_1, A_2 \dots A_K\}$$

dei suoi attributi, la proiezione di R su A è la

relazione di grado K che si ottiene considerando solo le colonne di R relative agli attributi contenuti in A ed eliminando le t-uple duplicate.

$$\pi_{A_1, A_2, \dots, A_K}(R) = \{t[A_1, A_2, \dots, A_K] | t \in R\}$$

### Restrizione

Data una relazione R ed un predicato P, la restrizione restituisce tutte le t-uple di R che soddisfano P.

$$P(t) = \{Classe = 5A1\}$$
$$\delta_P(R) = \{t | t \in R \text{ AND } P(t)\}$$

### Prodotto cartesiano

Restituisce l'insieme di tutte le possibili t-uple ottenute concatenando ogni t-upla della relazione R con ogni t-upla della relazione T.

$$r \text{ conc } s = (a_1, a_2 \dots a_n, b_1, b_2 \dots b_n)$$
$$R \times S = \{t | t = r \text{ conc } s, r \in R, s \in S\}$$

### Giunzione naturale

L'operazione di giunzione naturale (join) di R e S su un attributo A di R e un attributo B di S dello stesso tipo, restituisce una relazione di grado  $(g_1+g_2-1)$  le cui t-uple si ottengono con il seguente procedimento:

1. si calcola il prodotto cartesiano  $R \times S$ ;
2. si applica una restrizione sulle t-uple aventi gli attributi A e B dello stesso valore;
3. la relazione così ottenuta ha le colonne A e B uguali, per cui se ne elimina una delle due.

$$\begin{aligned} R \bowtie S \\ A = B \end{aligned}$$

Per la trattazione degli altri tipi di join, si rimanda al seguente link:

<http://www.dmi.unict.it/~giugno/database/ModelloEAlgebra%20Relazionale.pdf>

Applicando e componendo questi operatori, è possibile interrogare ed estrapolare dati dal database.

L'ultima fase di **progettazione** è quella **fisica**: a partire dallo schema logico definisco gli aspetti fisici della memorizzazione e rappresentazione in memoria di massa.

Per implementare al computer le basi di dati si utilizzano i **DBMS (DataBase Management System)**. Un DBMS è un insieme di strumenti-software che, sulla base delle specifiche dell'utente è in grado di gestire dati strutturati che sono tanti, importanti, condivisi, sia interrogati che aggiornati. Provvede inoltre alla generazione dello schema, alla ricerca e all'aggiornamento di dati. Nel caso specifico dei database relazionali si parla di RDBMS. Gli utenti di un DBMS vedono una macchina astratta che consente loro di compiere specifiche operazioni per descrivere e manipolare i dati contenuti nel database. Sul mercato sono disponibili vari prodotti software (complessi), quali ad esempio: Access, Oracle e SQLServer.

Un DBMS si può pensare come suddiviso in tre livelli di astrazione; dall'alto verso il basso troviamo:

- livello esterno: è quello con cui interagiscono i singoli utenti tramite viste (sottoinsiemi del DB);
- livello logico: troviamo gli schemi, le strutture logiche astratte del database;
- livello fisico: troviamo l'implementazione del database logico su file gestiti dal DBMS.

Un DBMS garantisce l'indipendenza fisica e logica: posso alterare i dati senza che venga alterata la struttura degli stessi.

Le interrogazioni (query) realizzate mediante la composizione degli operatori relazionali, vengono codificate nel **linguaggio SQL** (Structured Query Language). Si tratta di un linguaggio di tipo non procedurale utilizzato per creare, manipolare e interrogare database relazionali. L'SQL non è case sensitive (non distingue le lettere maiuscole dalle minuscole) e le istruzioni vengono separate

usando il punto e virgola “;”. Le istruzioni del linguaggio SQL si possono suddividere in tre macrocategorie:

- DML (data manipulation language): istruzioni per manipolare i dati contenuti nelle diverse tabelle (es. interrogazioni, inserimenti, cancellazioni, modifiche)
- DDL (data definition language): istruzioni per definire la struttura delle relazioni della base di dati (es. creazione e alterazione di tabelle e vincoli)
- DCL (data control language): istruzioni per controllare il modo in cui le operazioni vengono eseguite e per impostare autorizzazioni e permessi su singoli utenti

L'SQL può essere utilizzato in modalità stand-alone, ovvero interagendo direttamente con l'interprete SQL, o in modalità embedded, cioè costruendo ed eseguendo query SQL all'interno di un linguaggio ospite che presenta il supporto alla connessione con i database, ad esempio: PHP, ASP, JSP, Java, C/C++.

Vi sono moltissimi tutorial reperibili in rete per apprendere l'SQL, un buon punto di partenza può essere questo: <http://it.wikipedia.org/wiki/SQL>.

## CONCLUSIONI

In questo contributo abbiamo sintetizzato la trattazione delle basi di dati, modulo di fondamentale importanza del corso d'informatica nel secondo biennio superiore dell'istituto tecnico economico articolazione sistemi informativi aziendali (ex corso sperimentale programmatori mercurio). Tale trattazione consente agli studenti di acquisire alti livelli di competenze informatiche per la realizzazione dei database, ma, come abbiamo evidenziato, i modelli su cui si fondano queste applicazioni essenziali in tutti i settori della vita moderna, sono teorie e operatori matematici elementari.