Corso di studi in Biologia Molecolare
anno 2011-2012

Informatica e Bioinformatica

Insegnamento di Bioinformatica

Prof. Ivano Zara
E-mail: ivano.zara@unipd.it
Dipartimento di Biologia (VI piano sud)

Didattica di supporto:
Dott.ssa Chiara Gardin

Docenti per la parte Informatica:

- Prof. Giovanni Da San Martino

- Prof. Ivilin Stoianov (http://www.stoianov.it/)

Calendario lezioni
Lunedì 7 e 14 maggio dalle 12.30 alle 13.15 Martedì dalle 9.30 alle 11.15 aula G PR
Tot. 8 lezioni + 1 finale svolgimento esercizi d'esame

LABORATORIO BIOINFORMATICO

Giorno	Turno	Esercitazione
Martedì 15 maggio 14.15 - 18.15	I turno	Esercitazione 1 Aula CIV (Vallisneri)
Mercoledì 16 maggio 14.15 - 18.15	II turno	
Martedì 22 maggio 14.15 - 18.15	I turno	Esercitazione 2 Aula CIV (Vallisneri)
Mercoledì 23 maggio 14.15 - 18.15	II turno	
Martedì 29 maggio 14.15 - 18.15	Turno unico	Esercitazione 3 Aula Paolotti
Martedì 5 giugno 14.15 - 18.15	Turno unico	Esercitazione 4 Aula Paolotti

Premessa

- É gradita una minima conoscenza di Internet (ma in ogni caso si fa presto ad imparare!)
- Saper cercare le risorse in rete (sapete tutti cosa sono i motori di ricerca, es. Google?)
- I siti sono tutti in inglese …

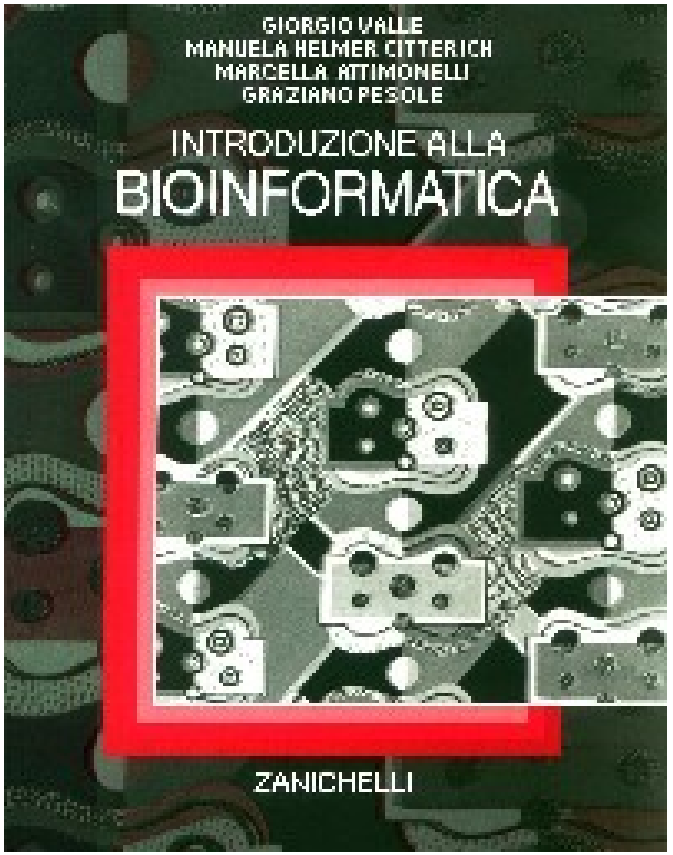
- **Supporto didattico.**
- Ho preparato queste presentazioni PPT che sono più che sufficienti per introdurvi alla materia. La rete è poi una vera e propria "miniera" di materiale (articoli, parti di libri, tutorial, corsi,…) per cui potrete sempre trovare quello che cercate.
- Credo che il provare personalmente sia il migliore supporto …

Le lezioni e le esercitazioni saranno inserite in:
http://didattica.cribi.unipd.it/bioinfouno

PER CHI VOLESSE SAPERNE DI PIÙ…

- A chi fosse interessato ad approfondire gli argomenti (per propri interessi personali, per la carriera,…) posso consigliare un libro.
- PRO: tutto il materiale è già organizzato ordinatamente (pappa pronta!)

- CONTRO: in un settore come questo, in rapidissima evoluzione, quando un libro esce è già vecchio! Il libro poi costa mentre Internet è gratis



Due siti interessanti per reperire informazioni e strumenti utili per la bioinformatica:

- '**2can Bioinformatic Support Portal**' **The bioinformatics educational resource** presente all'EBI (European Bioinformatics Institute) (**http://www.ebi.ac.uk/2can/home.html)**

- Basic Introduction to the Science Underlying NCBI Resources. Presente all'NCBI (National Center for Biotechnology Information) (**http://www.ncbi.nlm.nih.gov/About/primer/bioinformatics.html)**

COS'É LA BIOINFORMATICA?

"Applicazione dell'informatica alla gestione e all'analisi dei dati biologici"

Bioinformatics is an interdisciplinary research area that is the interface between the biological and computational sciences. The ultimate goal of bioinformatics is to uncover the wealth of biological information hidden in the mass of data and obtain a clearer insight into the fundamental biology of organisms. This new knowledge could have profound impacts on fields as varied as human health, agriculture, the environment, energy and biotechnology.

COSA HA DETERMINATO LO SVILUPPO DELLA BIOINFORMATICA:

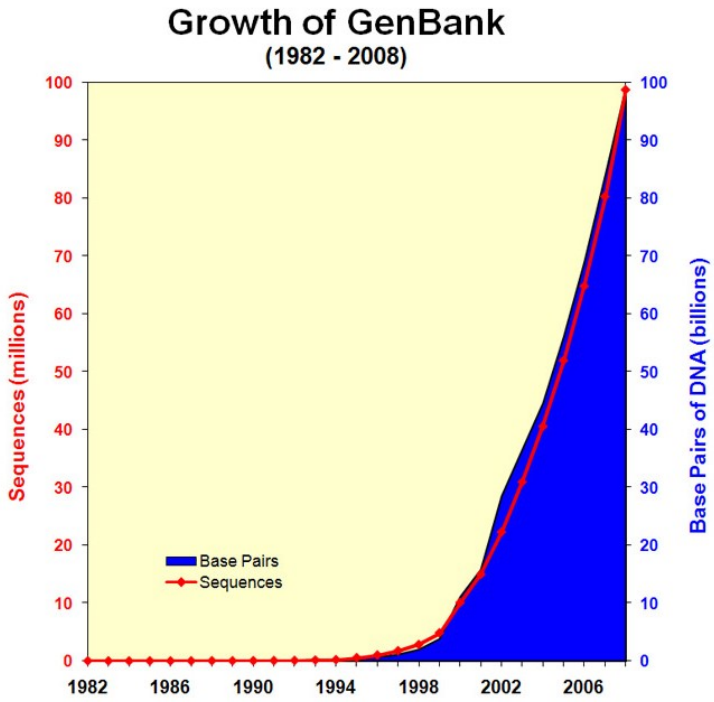
- Sviluppo di biotecnologie innovative
- Sviluppo delle potenzialità informatiche (hardware)
- Sviluppo di nuovi programmi informatici (software)
- Diffusione delle conoscenze informatiche tra i biologi

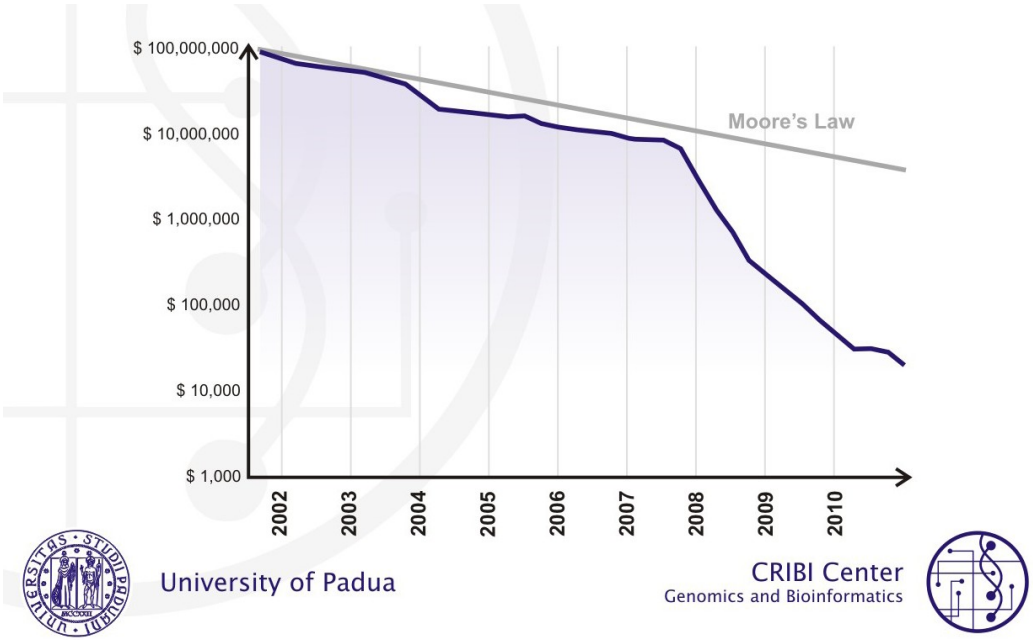
I “DATI” BIOLOGICI…

Negli ultimi 20-30 anni abbiamo assistito ad una vera e propria esplosione nella produzione di dati biologici

(aumento esponenziale)

Costo sequenze / costo hardware





La legge di Moore: ogni 18-24 mesi raddoppia il numero dei transistor contenuti nei circuiti integrati (http://www.intel.com/technology/mooreslaw/)

Per una bioinformatica 'di base' è necessario:

-Sapere cosa sono e come sono strutturati i database

-Avere conoscenze biologiche

-Conoscere **dove** sono archiviati i dati biologici

-Conoscere **come** sono archiviati questi dati

-Saper effettuare ricerche (anche complesse)

-Essere in grado di utilizzare i numerosi strumenti ('tool') che sono pubblicamente disponibili

E' necessario una vostra conoscenza (di base) di alcuni argomenti biologici (DNA, proteine, gene, genoma, procariote, eucariote).

Dovreste già conoscerli (dalle superiori) e comunque studiati durante il corso di Biochimica.

Comunque, durante il corso, cercheremo di ovviare ad eventuali vostre lacune

COSA TRATTEREMO?

Si tratta di un corso breve, di otto ore frontali e 4 esercitazioni di 4 ore, che hanno lo scopo di introdurre sommariamente alcuni dei principali argomenti della biologia e di fornire gli strumenti ed i metodi per accedere all'informazione biologica in modo razionale ed efficiente, utilizzando le risorse disponibili in rete.

Possiamo dividere il corso in tre parti:

A) Archiviazione dati: DATABASE
- come vengono memorizzati i dati
- come strutturare gli archivi

B) Database Biologici (ne esistono molti tipi, vedremo i più importanti)
- database di sequenze di DNA e proteine
- database articoli scientifici
- database malattie genetiche, mutazioni ecc.

C) Analisi computazionale dei dati. In particolare:
- allineamento di sequenze e ricerca di similarità
- Uso di strumenti informatici (tools) per analizzare dati

A) Database

Archivio dati

Prima di parlare di database introduciamo gli archivi dati

Esistono molte informazioni e molti dati → c'è la necessità di memorizzarli e conservarli

Prima dell'avvento del PC le informazioni venivano memorizzate su supporti fisici quali la carta

Come? → predisponendo apposite strutture di conservazione dei dati

- Semplici strutture quali registri o quaderni di appunti → *la memorizzazione dei dati è sequenziale e non permette un ordinamento specifico*
- Sistema più evoluto: **Schedario**

L'elemento principale è la scheda che caratterizza ogni elemento dell'archivio

Un **archivio** (schedario) non può contenere 'tutto' ; deve essere costruito per lo scopo a cui serve

Esempi:

-Si vuole memorizzare i modelli di automobili → lo schedario dovrà essere costituito da una scheda per ogni modello di automobile

-Si vuole costruire uno schedario clienti → ad ogni cliente dovrà essere associata una scheda

La **scheda** è l'elemento principale dello schedario, ma quello che caratterizza l'archivio è il contenuto delle schede

Ogni scheda deve contenere le informazioni cioè gli **attributi** (chiamati anche categorie di informazioni) che caratterizzano l'elemento

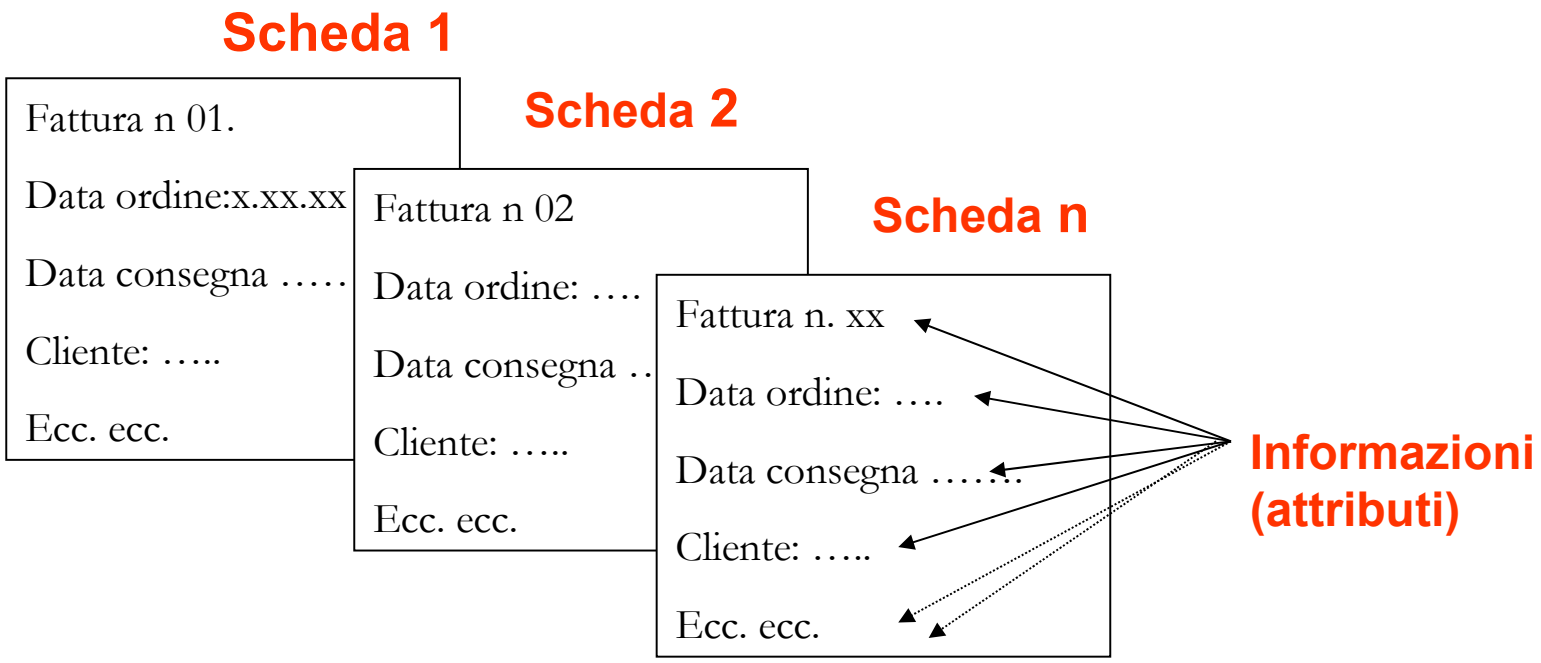
Esempi:

Archivio modelli auto: gli attributi contenuti nelle schede potrebbero essere: nome del modello, marca, anno di fabbricazione, motorizzazione, ecc.

Archivio per registrare le vendite dovrà essere caratterizzato dalle informazione 'cliente', 'fatture', 'prodotti'

Esempio: **schedario di fatturazioni di una ditta commerciale**.
Ogni fattura è rappresentata da una (e una sola) scheda ognuna di questa deve contenere le informazioni correlate (attributi).
Esempi di attributi possibili: data dell'ordine, data della consegna, importo, nome del cliente, indirizzo del cliente, telefono del cliente, ecc...

Notare che le informazioni contenute su una scheda possono anche essere ripetute su altre schede (uno stesso cliente può essere associato a differenti fatture, differenti fatture possono essere emesse nello stesso giorno, **la fattura però deve essere univoca e rappresentare solo quella specifica transizione di merce.**



DATABASE

In informatica, il termine **database**, tradotto in italiano con **banca dati**, **base di dati** o anche **base dati**, indica un archivio di dati, riguardanti uno stesso argomento o più argomenti correlati tra loro, strutturato in modo tale da consentire la gestione dei dati stessi (l' inserimento, la ricerca, la cancellazione ed il loro aggiornamento) da parte di applicazioni software gestite da un elaboratore

Altre definizioni:

- È l'archivio di dati eterogenei gestito da un elaboratore in grado di memorizzare e organizzare i dati per velocizzarne la gestione (ad esempio, inserimento nuovi dati, modifica dati esistenti, ricerca dati).

-Un database e' un archivio, solitamente un file, contenente una struttura di dati correlati.

- Un Database può essere definito come un insieme di informazioni strettamente correlate e memorizzate su un supporto di memoria di massa, costituenti un tutt'uno, che possono essere manipolate da più programmi applicativi.

-Definizione più completa: Si definiscono **strutture dati** i modi di organizzare secondo regole precise una certa quantità, detta anche "base", di informazioni; tali regole possono definirsi sia in forma teorica, sia in forma pratica, riferendo quest'ultima all'effettivo ordinamento fisico dei dati sulla memoria di un elaboratore elettronico, la parte di memoria di un elaboratore elettronico destinato a contenere le informazioni così organizzate prende il nome di **database**.

ORGANIZZAZIONE dei Database

L'organizzazione dei dati in un database è simile a quella di uno schedario.
Sono organizzati in schede contenenti specifici attributi che caratterizzano le schede e quindi l'archivio
Le uniche differenze sono:

a) Le definizioni:

- le schede prendono il nome di **RECORDS**
- gli attributi prendono il nome di **CAMPI (FIELDS)**

b) il modo con cui vengono memorizzate (gli schedari su supporto cartaceo, i database nelle memorie fisiche dei computer);

c) La gestione dei dati (manuale negli schedari) gestita da software negli elaboratori.

Sono evidenti i vantaggi dei database rispetto agli schedari

- La ricerca di schede (record) è facilitata in quanto gestita da appositi software

- Si possono facilmente estrarre un insieme di record caratterizzati da particolari dati.

- Si possono ottenere velocemente liste di record ordinati per argomenti specifici (esempio per data, ordine alfabetico ecc.)

- ecc. ecc.

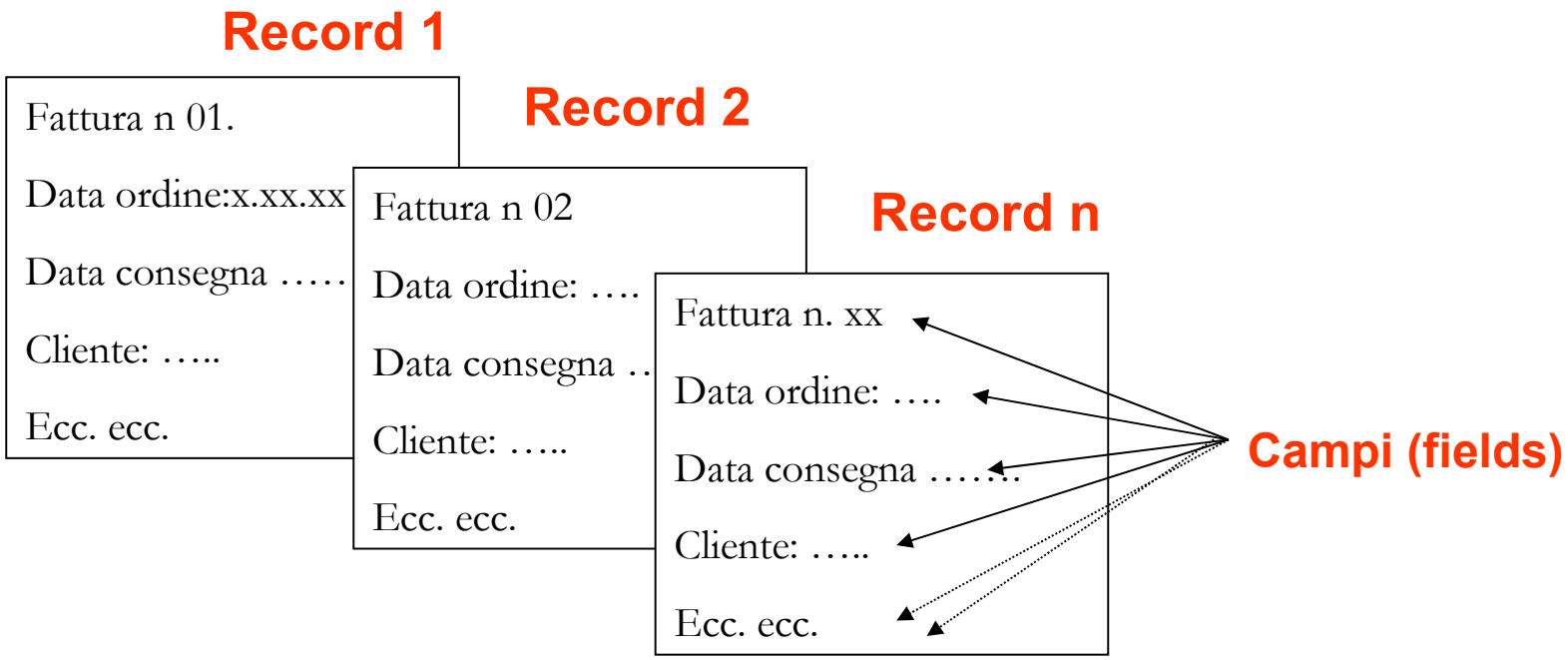
Rappresentazione di una Struttura dei database

(simile allo schedario visto in precedenza)

Records e campi di un database

Un modo semplice di immaginare un database è di pensare ad uno schedario di fatturazioni di una ditta commerciale.

Ogni fattura è rappresentata da una scheda (**record**), ognuna di questa deve contenere una serie di dati (**campi**), ad esempio: data dell'ordine, data della consegna, importo, nome del cliente, indirizzo del cliente, telefono del cliente, ecc...



Nei database valgono le stesse proprietà degli schedari

Se il database rappresenta un'entità del mondo reale, allora **ogni record rappresenta un'istanza di quella entità** e quindi non possono esistere più record per la stessa istanza.

Da questo deriva che un database è costituito da tanti record (le schede) ognuno dei quali rappresenta una sola istanza dell'insieme riportato del DB

(Esempio se definiamo un database di sequenze di proteine, avremo un record per ogni sequenza proteica).

Ogni campo rappresenta invece un attributo dell'entità da riprodurre

(esempio database di automobili: marca, modello, cilindrata ecc)

Identificatore di record

È importante avere un contrassegno che identifica in modo univoco il record. Deve quindi esistere un **campo speciale** chiamato '**chiave**' che deve essere diverso per ogni record. Il campo chiave può essere rappresentato da un numero progressivo, oppure da una sigla, o anche da un nome, comunque sia, è essenziale che sia unico. Molto spesso il campo chiave viene chiamato "**ID**". (In alcuni database biologici viene chiamato "AC" (Accession number))

Metodi per memorizzati i dati nei database

Esistono essenzialmente due modi differenti per la gestione dei dati in un database:

- Database flat file:

Tutti i dati sono memorizzati in un unico file.

Il file può essere strutturato in due modi differenti

formato testuale

formato tabella

-Database relazionale

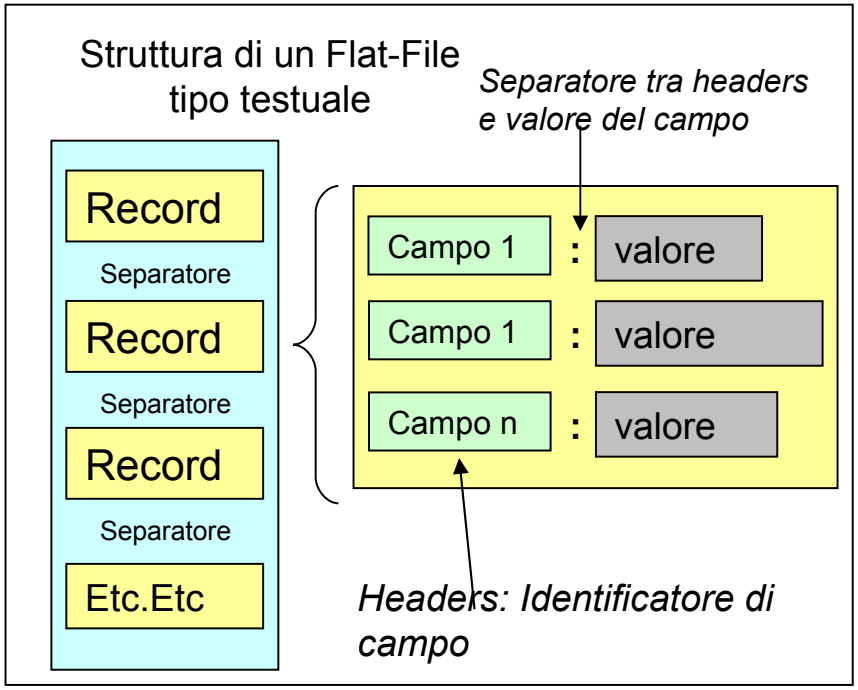
-I dati sono memorizzati in più file collegati tra loro (vedremo più avanti)

-Per la loro gestione, sono necessari appositi software

Database tipo FLAT-FILE in formato testuale

Un database può essere memorizzato semplicemente in un file di testo: in ogni byte è memorizzato un carattere (esplicito o speciale).

Tutti i record sono scritti in modo seriale, separati tra loro da particolari spaziatori (caratteri speciali), con i relativi dati (campi) scritti al loro interno separati tra di loro da altri spaziatori



I contenuti di un record devono essere inequivocabilmente identificabili.
Servono, quindi, degli identificatori di campo
Questi identificatori possono corrispondere semplicemente alla posizione (es. primo campo, secondo campo, ecc.)
oppure possono essere delle "etichette-intestazioni" (**headers**) che indicano il nome del campo, seguite da un 'separatore' e poi dal valore che assume il campo in quel record.

Nei flat-file tipo testuale possono esistere più campi con lo stesso nome.

(esempio in un database di prodotti, in cui ogni record riporta le caratteristiche di un prodotto, possono esserci più avvertenze per lo stesso prodotto).

In un database di molecole biologiche, possono esserci più campi che riportano articoli scientifici inerenti la molecola

UN ESEMPIO

Quanti record sono? (quale è il separatore di record?)

Quali campi ha ciascun record? (quali sono i separatori e gli identificatori dei record?)

Quale è il campo chiave? Che valori assume?

Osservare la presenza di più campi con lo stesso nome nello stesso record

Osservare che alcuni campi possono mancare in alcuni record

```
ID                : 28877
PARENT ID         : 28876
RANK              : no rank
GC ID             : 1
SCIENTIFIC NAME   : IDIR agent
SYNONYM           : Infectious Disease of Infant Rats
SYNONYM           : Rotavirus (GROUP B / STRAIN IDIR)
SYNONYM           : infectious diarrhea of infant rats agent IDIR
//
ID                : 55279
PARENT ID         : 6607
RANK              : family
GC ID             : 1
MGC ID            : 5
SCIENTIFIC NAME   : Idiosepiidae
//
ID                : 82764
PARENT ID         : 82761
RANK              : family
GC ID             : 1
MGC ID            : 5
SCIENTIFIC NAME   : Idoteidae
//
```

Sempre più difficile ……

```
MGI:11945 Ablim1    actin-binding LIM protein     GDB:7173461    ABLIM1     3983
MGI:87902 Acta1     actin, alpha 1, skeletal muscle    GDB:120535     ACTA1     58
MGI:87909 Acta2     actin, alpha 2, smooth muscle, aorta    GDB:125197     ACTA2     59
MGI:87904 Actb actin, beta, cytoplasmic GDB:118964     ACTB 60
```

Quale è il separatore di record? Quale è l'identificatore (separatore) di campo?

Rappresentiamo anche i caratteri nascosti

```
MGI:11945→Ablim1 → actin-binding·LIM·protein → GDB:7173461 → ABLIM1 → 3983¶
MGI:87902→Acta1 → actin,·alpha·1,·skeletal·muscle → GDB:120535 → ACTA1 → 58¶
MGI:87909→Acta2 → actin,·alpha·2,·smooth·muscle,·aorta → GDB:125197 → ACTA2 → 59¶
MGI:87904→Actb→actin,·beta,·cytoplasmic→GDB:118964 → ACTB→60¶
```

Le righe (record) sono separate dal carattere NEW-LINE (vai a capo) " ¶ "
I campi sono separati dal carattere TAB (tabulazione) " " →

Notate che in questo caso il campo è individuato solo dalle posizione assunta nella riga (non esiste un nome o un codice che lo contraddistingue). E' quindi obbligatorio, anche per i campi vuoti, scrivere i relativi separatori di campo.

Esempio di UN RECORD BIOLOGICO
(tipo flat file formato testo)

Campi evidenziati:

-LOCUS: un codice
-DEFINITION: descrizione dell'entry
-ACCESSION un codice (campo chiave corrisponde all'ID)
-ORGANISM: l'organismo a cui appartiene la sequenza (e tassonomia)
-REFERENCE: Riferimenti bibliografici a quella sequenza o chi l'ha sottomessa
-FEATURES: alcune caratteristiche e link importanti
-ORIGIN: la sequenza

NCBI Nucleotide

Entrez PubMed Nucleotide Protein Genome Structure PMC Taxonomy Books

Search Nucleotide for Go Clear

Limits Preview/Index History Clipboard Details

Display default Show: 20 Send to File Get Subsequence Features

1: AY536527. Sus scrofa growth...[gi:46361728]

```
LOCUS       AY536527                658 bp    mRNA    linear   MAM 18-APR-2004
DEFINITION  Sus scrofa growth hormone mRNA, complete cds.
ACCESSION   AY536527
VERSION     AY536527.1  GI:46361728
KEYWORDS    .
SOURCE      Sus scrofa (pig)
  ORGANISM  Sus scrofa
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Cetartiodactyla; Suina; Suidae; Sus.
REFERENCE   1  (bases 1 to 658)
  AUTHORS   Feng,J., Li,W.-F. and Xu,Z.-R.
  TITLE     Direct Submission
  JOURNAL   Submitted (31-JAN-2004) Zhejiang University, Feed Science
            Institute, 164 Quitao North Road, Hangzhou, Zhejiang 310029, China
FEATURES             Location/Qualifiers
     source          1..658
                     /organism="Sus scrofa"
                     /mol_type="mRNA"
                     /db_xref="taxon:9823"
     CDS             8..658
                     /codon_start=1
                     /product="growth hormone"
                     /protein_id="AAS89356.1"
                     /db_xref="GI:46361729"
                     /translation="MAAGPRTSVLLAFALLCLPWTQEVGAFPAMPLSSLFANAVLRAQ
                     HLHQLAADTYKEFERAYIPEGQRYSIQNAQAAFCFSETIPAPTGKDEAQQRSDVELLR
                     FSLLLIQSWLGPVQFLSRVFTNSLVFGTSDRVYEKLKDLEEGIQALMRELEDGSPRAG
                     QILKQTYDKFDTNLRSDDALLKNYGLLSCFKKDLHKAETYLRVMKCRRFVESSCAF"
ORIGIN
        1 ggctgtgatg gctgcaggcc ctcggacctc cgtgctcctg gctttcgccc tgctctgcct
       61 gccctggact caggaggtgg gagccttccc agccatgccc ttgtccagcc tatttgccaa
      121 cgccgtgctc cgggcccagc acctgcacca actggctgcc gacacctaca aggagtttga
      181 gcgcgcctac atcccggagg gacagaggta ctccatccag aacgcccagg ctgccttctg
      241 cttctcggag accatcccag cccccacggg caaggacgag gcccagcaga gatcggacgt
      301 ggagctgctg cgcttctcgc tgctgctcat ccagtcgtgg ctcgggcccg tgcagttcct
      361 cagcagggtc ttcaccaaca gcctggtgtt tggcacctca gaccgcgtct acgagaagct
      421 gaaggacctg gaggagggca tccaggccct gatgcgggag ctggaggatg gcagcccccg
      481 ggcaggacag atcctcaagc aaacctacga caaatttgac acaaacttgc gcagtgatga
      541 cgcgctgctt aagaactacg ggctgctctc ctgcttcaag aaggacctgc acaaggctga
      601 gacatacctg cgggtcatga agtgtcgccg cttcgtggag agcagctgtg ccttctag
//
```

Disclaimer | Write to the Help Desk
NCBI | NLM | NIH

Un database flat-file può anche essere memorizzato in formato tabella

Campi

Records

ID	Marca	Modello	Potenza (KW)	Consumo (km/l)	Costo (euro)	Ecc. ecc.
01	FIAT	Panda	40	17.5	8600	---
02	Ferrari	F430	360	5.4	151000	---
03	Mercedes	Classe A	70	16.1	18000	---
04	Citroen	C3 Cabrio	54	14.7	15000	---
Ecc ecc	---	---	---	---	---	---

In questo caso, ogni riga rappresenta un record e ogni colonna rappresenta un campo. I nomi delle colonne rappresentano i nomi dei campi dei record.

Solitamente, in questi tipi di database, un campo può contenere valori solo di un certo tipo (numeri interi, numeri reali, date, stringhe di caratteri,…)

In una tabella, tutti i record hanno tutti gli stessi campi (corrispondono alle colonne). Se in un record, non viene definito un campo, si dovrà usare un valore speciale 'NULL' per riempirlo (nei flat-file invece determinati campi potevano tranquillamente essere non definiti per un record).

Questo tipo di database è utilizzato soprattutto per costruire i database relazionali (che vedremo più avanti)

Confronto DB tipo testuale vs DB tipo tabella

I **DB formato tabelle** sono strutture chiuse, tutti i record devono essere già preformattati e predisposti per il numero massimo di campi che il record può contenere (se un campo è vuoto, deve comunque essere riempito con un carattere 'nullo'). In positivo: i record sono più facilmente confrontabili e servono per creare DB di tipo relazionale.

Nei **DB formato testuale** i record possono ospitare un numero variabile di campi, se un campo è vuoto, non serve dichiararlo.
- possono esistere campi multipli con lo stesso nome;
- possono esistere dei 'sottocampi' (un campo può essere considerato un record contenente dei propri campi)

Esempio di un record di un DB flat-file testuale con la presenza di 'campi secondari'

In questo caso, un campo può essere visto come un record contenente a sua volta propri campi

'REFERENCE' Campo 'principale'

Campi 'secondari' (all'interno del campo 'REFERENCE')

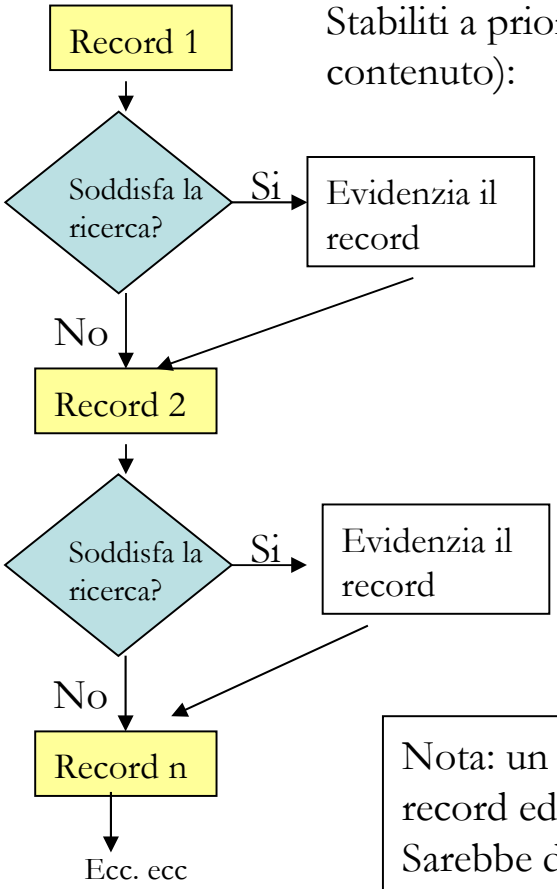
```
LOCUS       JU323950               1371 bp    mRNA    linear   TSA 26-MAR-2012
DEFINITION  TSA: Macaca mulatta Mamu_517051 mRNA sequence.
ACCESSION   JU323950
VERSION     JU323950.1  GI:380791660
DBLINK      BioProject: PRJNA77627
            Sequence Read Archive: SRR358985
KEYWORDS    TSA; Transcriptome Shotgun Assembly.
SOURCE      Macaca mulatta (Rhesus monkey)
  ORGANISM  Macaca mulatta
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;
            Catarrhini; Cercopithecidae; Cercopithecinae; Macaca.
REFERENCE   1  (bases 1 to 1371)
  AUTHORS   Pandey,S., Maudhoo,M.D., Guda,C., Ferguson,B., Fox,H. and
            Norgren,R.B.
  TITLE     De novo assembly of the rhesus macaque transcriptome from NextGen
            mRNA sequences
  JOURNAL   Unpublished
REFERENCE   2  (bases 1 to 1371)
  AUTHORS   Pandey,S., Maudhoo,M.D., Guda,C., Ferguson,B., Fox,H. and
            Norgren,R.B.
  TITLE     Direct Submission
  JOURNAL   Submitted (27-FEB-2012) Genetics, Cell Biology and Anatomy,
            University of Nebraska Medical Center, 985805 Nebraska Medical
            Center, Omaha, NE 68198-5805, USA
COMMENT     All reads were aligned with the human RefSeq mRNA sequences using
```

Ordine ed estrazione dati da un database

Nei sistemi 'statici', gli archivi sono creati in modo cronologico.
Ogni nuovo record viene aggiunto e memorizzato alla fine del relativo file.
Si dice che **il database è ordinato secondo la data di inserzione dei record**

Algoritmo per la scansione di un DB alla ricerca di particolari record

Stabiliti a priori i parametri che devono soddisfare la ricerca (campi con il loro contenuto):



Il relativo software deve analizzare un record alla volta, verificando il contenuto dei campi cercati e, in caso affermativo, selezionare e/o visualizzare il record.

Nota: un file di dati biologico può contenere migliaia o milioni di record ed ognuno di questi può contenere molteplici campi. Sarebbe dispendioso in termini di tempo e risorse scandire tutto il file alla ricerca di un particolare record. → Necessità di utilizzare nuove strategie come **ordinare i database**

Database ordinati

Cosa significa ordinare un database ?

→ Mettere in ordine fisicamente o virtualmente (vedremo fra poco cosa significa virtualmente) i record secondo il contenuto di uno o più particolari campi.

Ad esempio in un database di studenti universitari, si potrebbe creare una copia del DB ordinato secondo i campi 'cognome' e 'nome'. Questo faciliterebbe la ricerca di un particolare studente.

Creando, invece, una copia del DB ordinato secondo il campo 'facoltà": gli studenti sarebbero ragruppati secondo la facoltà di iscrizione. Questo faciliterebbe l'estrazione dei reord relativi agli studenti di una particolare facoltà.

Un archivio ordinato permette l'applicazione di particolari algoritmi che velocizzano la ricerca esempio:

Algoritmo di ricerca dicotomica (detta anche ricerca binaria)

ad ogni passaggio, si analizza il record che si trova in posizione centrale, se il record risulta quello voluto → fine della ricerca, altrimenti, essendo il DB ordinato, si individua facilmente se il record cercato si trova a monte o a valle, e si scarta la parte contenente la metà dei record che non contengono il dato voluto.

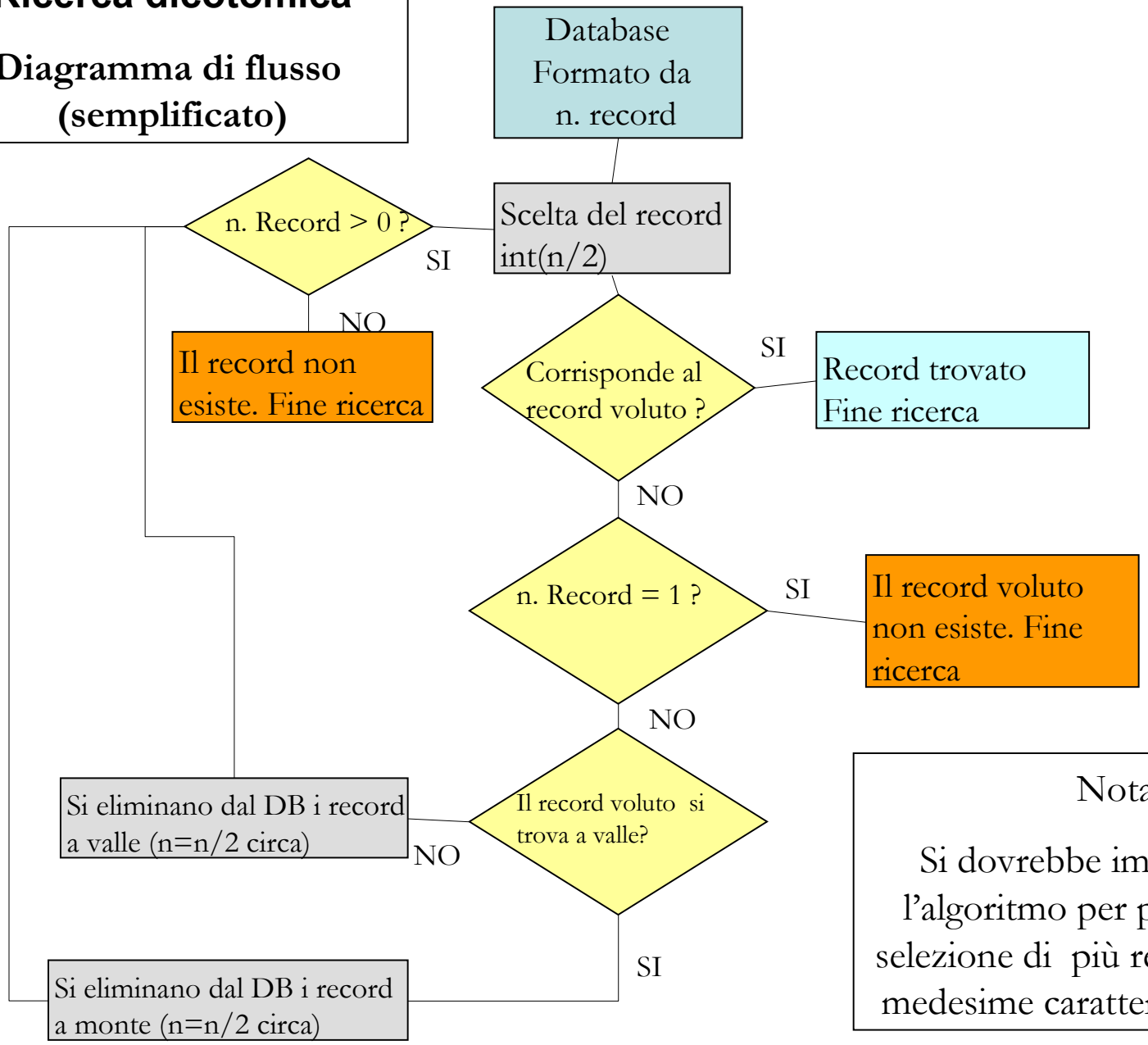
La ricerca continua (**reiterazione**) nello stesso modo, eliminando di volta in volta metà dell'elenco rimasto, fino ad arrivare all'ultimo confronto che ci darà l'informazione richiesta, o ……. fino a scoprire che l'elemento non si trova nel database

Es. per un DB, contenente 128 record, sono sufficienti al massimo 8 passaggi (reiterazioni) per individuare il record voluto (se esiste) (128 → 64 → 32 → 16→ 8 → 4 → 2 → 1 → 0)

(log2(128) +1) = 8)

Ricerca dicotomica

Diagramma di flusso (semplificato)



Nota:

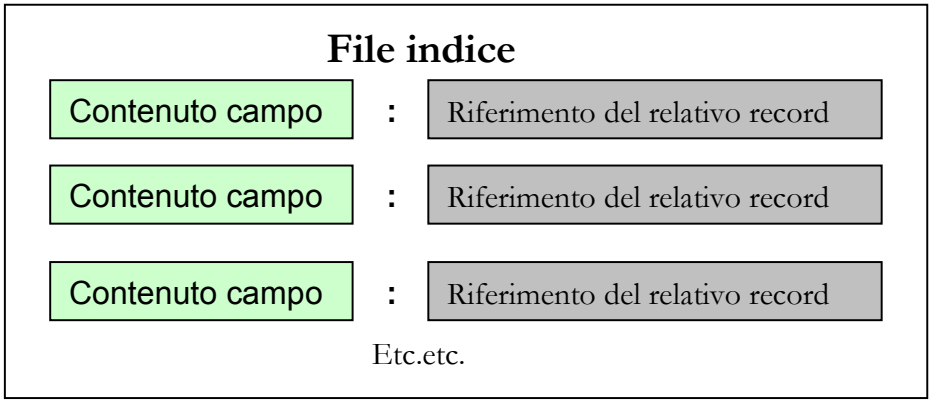
Si dovrebbe implementare l'algoritmo per permettere la selezione di più record aventi le medesime caratteristiche volute

Per applicare una ricerca dicotomica o applicare altri particolari algoritmi è necessario che il DB sia ordinato. Quindi è necessario disporre di un database ordinato per ogni tipo di ricerca (necessità di duplicare il database).

E' evidente che, se il database fosse di grandi dimensioni, servirebbe un enorme spazio fisico per memorizzare le differenti forme del DB (notare che ogni database ordinato occupa lo stesso spazio del DB originale) e servirebbero lunghi tempi di elaborazione per ordinarli (ogni volta che si modifica il DB si deve riscrivere tutti i DB ordinati correlati).

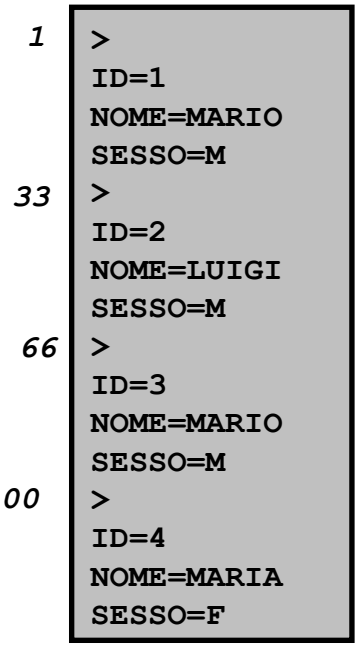
Nuova strategia → indicizzazione dei database

Invece di memorizzare e riscrivere in modo ordinato un DB, è sufficiente disporre di una lista che riporti la posizione dei record, ordinati secondo il tipo di ricerca da effettuare e memorizzata in un 'piccolo' file (**file indice**)



E' necessario creare una lista (indice) per ogni tipo di ricerca che si vuole effettuare (ordinata in modo opportuno)

ESEMPIO indicizzazione di un flat-file tipo testuale



Es. 2 Indicizzazione secondo il nome

NOME

```
LUIGI:33
MARIA:100
MARIO:1,66
```

Es. 3 Indicizzazione secondo il sesso

SESSO

```
F:100
M:1,33,66
```

Le ricerche sono effettuate solo sugli indici

risultano più veloci.
(gli indici sono ordinati)

Generalmente i record di un database testuale sono organizzati e memorizzati in file '*heap*' (catasta) in ordine casuale, le loro dimensioni possono essere tra loro differenti ed occupano quantità di memoria differenti

Nei file indici, anziché riportare i riferimenti numerici ai record ordinati, si preferisce riportare gli indirizzi di memoria ('puntatori') dove sono memorizzati i record nel supporto fisico.
Questo facilita e velocizza il posizionamento/ritrovamento dei record.

DATABASE RELAZIONALE

Nei DB flat-file, potrebbe essere necessario scrivere/memorizzare lo stesso dato in più record o anche in record di DB differenti → '**ridondanza dell'informazione**'.
Esempio in un DB di fatture: l'informazione relativa ad ogni cliente (nome, indirizzo, telefono, ecc...) deve essere ripetuta ogni volta che una fattura viene emessa per lo stesso cliente.

Inoltre, se un particolare dato dovesse cambiare, sarebbe necessario modificare tutti i record dove tale dato è archiviato.
Esempio: se il cliente cambiasse indirizzo allora si dovrebbe cambiare l'indirizzo a tutte le fatturazioni in sospeso.
Se non modificassimo l'indirizzo su tutti i record, non si potrebbe risalire con sicurezza all'esatto cliente. *Esempio come potremmo stabilire se il sig. Bianchi di via Verdi e il sig. Bianchi di via Rossini siano la stessa persona che ha cambiato abitazione oppure due persone diverse?*

Per evitare tali problematiche, è necessario che un *dato comune* venga scritto/memorizzato una sola volta, in un'apposita struttura (DB) . I record che utilizzeranno dati comuni potranno, semplicemente, contenere un riferimento alla struttura che contiene il relativo dato.

Nell'esempio ordini-fatturazioni, si potrebbe predisporre un DB in formato tabella dei **clienti** in cui ogni record corrisponde ad uno solo cliente in modo inequivocabile;
Poi creare un DB degli ordini in cui, invece di ripetere i dati del cliente, si riporterà solo il riferimento (esempio l'ID) al relativo record presente nella tabella clienti.
Un apposito software provvederà poi a legare i record di tabelle differenti creando così un database unico che viene detto **database relazionale**.

Un DATABASE RELAZIONALE è un insieme di DB (generalmente in formato tabella) correlati tra loro attraverso relazioni che fanno riferimento ai campi 'chiave'

DB clienti

ID cliente	Nome	Indirizzo	Ecc.ecc.
01	Rossi Guido	Padova, via ….	..
02	Barca Mario	Milano, ….	..
03	Milan Lino	….	..
ecc	..	..	..

DB prodotti

ID prodotto	Descrizione	Costo unitario	Ecc ecc
01	Armadio …..	550	---
02	Tavolo …..	100	---
Ecc ecc	---	---	---

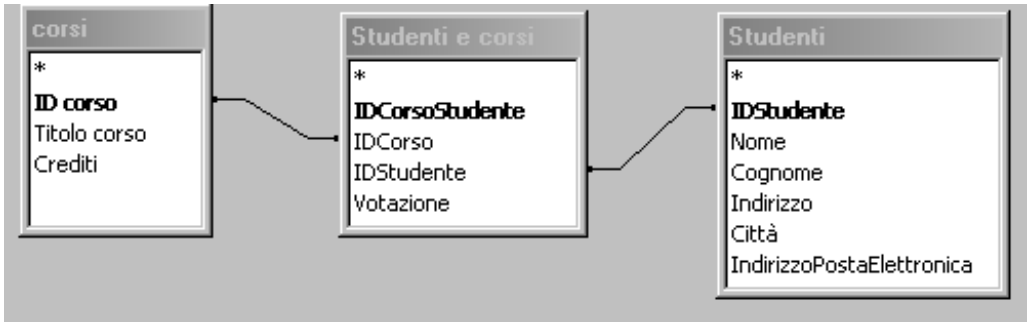
Relazione

Relazione

DB ordini

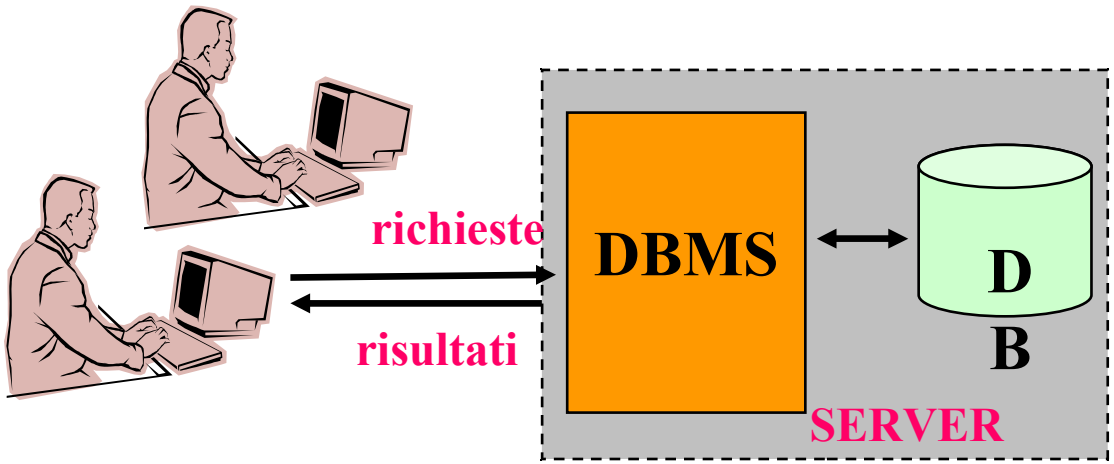
ID ordine	ID cliente	Data ordine	ID Prodotto	Ecc ecc
01	03	24/12/2005	02	---
02	---	---	---	---
---	---	---	---	---

Struttura (semplificata) di un DB relazionale studenti/corsi ottenuto con Access (Microsoft)



I database relazionali necessitano di particolari programmi di gestione *(Database Management System o DBMS)* , che siano in grado di saltare da una tabella all'altra e di capire le relazioni ed i vincoli ad esse associati. Devono inoltre occuparsi di gestire l'aggiunta, la modifica e la gestione degli indici

Il DBMS funge da interfaccia verso il database, in una tipica configurazione CLIENT-SERVER.
Il *server* è residente su un computer remoto, mentre i *client* sono in generale gli altri computer.



SQL Structured Query Language

- E' sicuramente il DBMS relazionale più diffuso.
- SQL è uno standard di cui esistono alcune implementazioni
- 

ORACLE (commerciale)
- 

MySQL (free) (lo imparerete al terzo anno)

Nota:

Al fine di rendere più semplice la consultazione dei dati, molti database biologici sono estratti dai loro sistemi di gestione relazionale e sono "appiattiti" in file di testo chiamati "***flat files***" leggibili come semplici file di testo. Ricordo che i "flat files" possono essere indicizzati e utilizzati per ricerche anche molto complesse (ENTREZ, SRS).