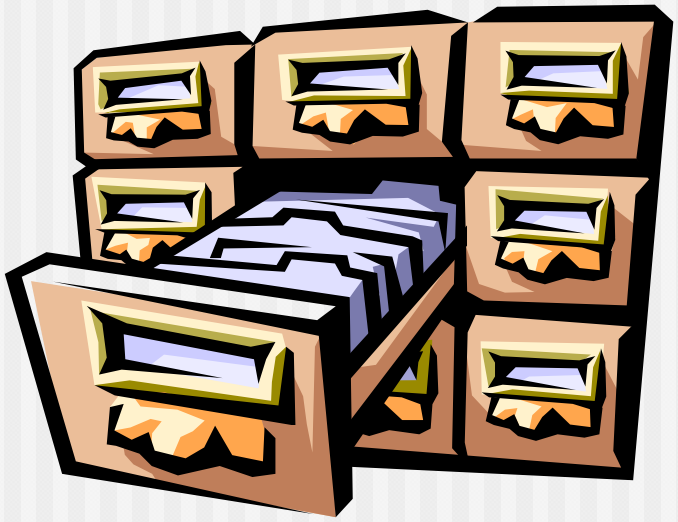
Le Strutture Dati Su Disco
Parte Prima



Dalle Strutture Dati ai Data Bases

Strutture dati in memoria centrale e su memoria di massa

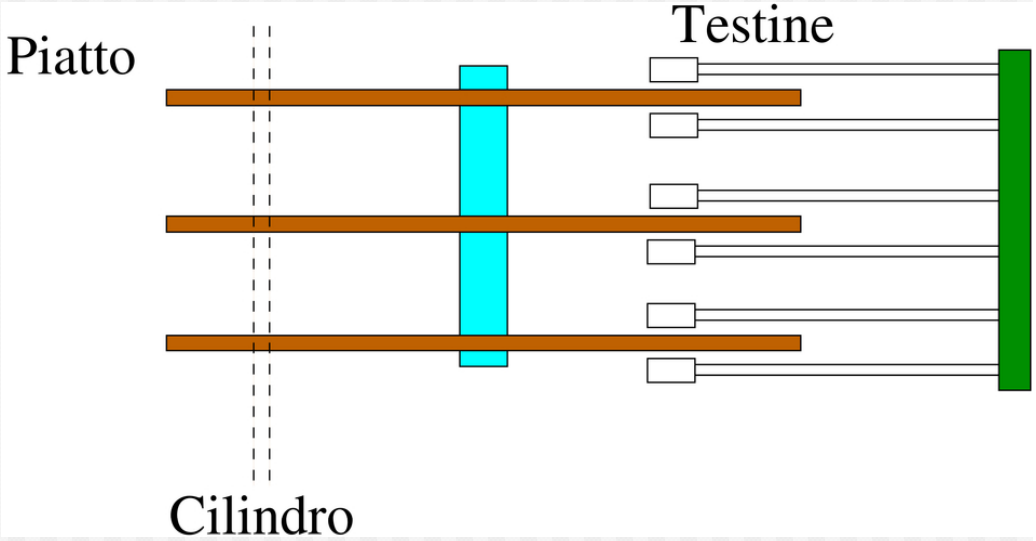
- Per comprendere le particolarità delle strutture dati in memoria di massa rispetto a quelle in memoria centrale, bisogna confrontare
  - le differenti organizzazioni fisiche
  - i differenti tempi di accesso
  - la volatilità della memoria centrale e la persistenza della memoria di massa

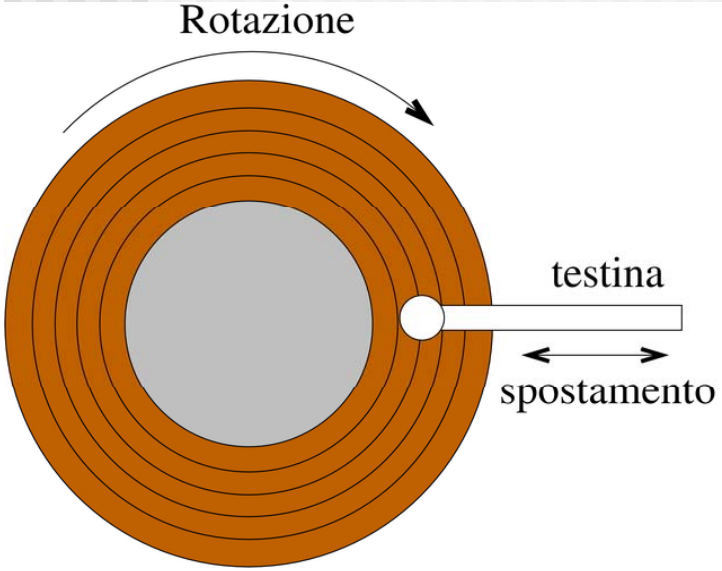
L’organizzazione della memoria centrale

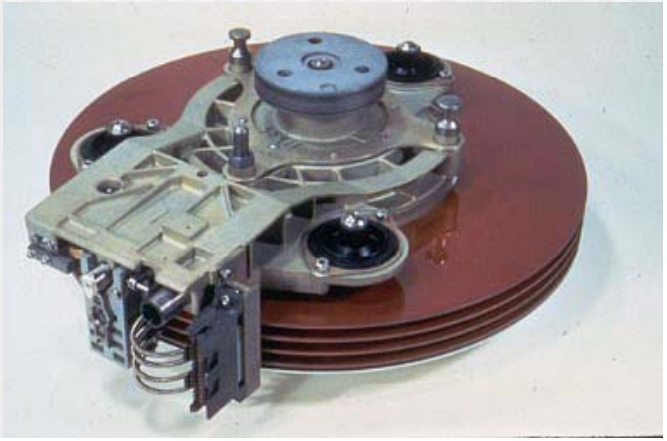
- Costituita da una successione lineare consecutiva di caratteri (bytes) di otto bit
- Ogni byte è accessibile con un indirizzo intero da zero ad un massimo n
- Il byte i+1 segue il byte i
- I tempi di accesso sono omogenei e costanti
- I tempi di accesso sono dell’ordine dei 100 nanosecondi (approssimando per semplificare i calcoli)

L’organizzazione della memoria di massa

- Costituita da piatti rotanti con velocità costante
- Ogni piatto di solito ha due testine

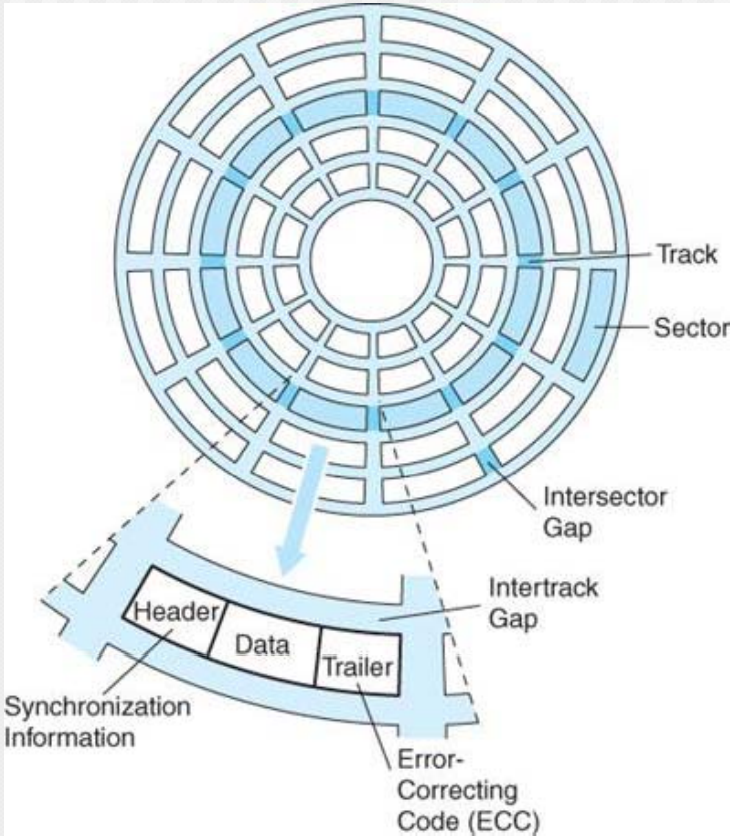






Tracce e settori

- Ogni piatto è diviso in tracce concentriche
- Ogni traccia è divisa in settori di dimensione costante
- Il settore è l'unità minima di lettura e scrittura
- Ogni settore è accessibile fornendo la terna (testina, traccia, settore)



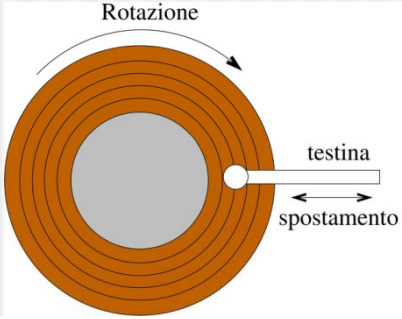
Tempi di accesso

- I tempi di accesso non sono costanti né omogenei, ma casuali, secondo la formula
- Tt = Tseek + Trot + Tlet
  - Tseek tempo di posizionamento della testina sulla traccia (tempo di ricerca)
  - Trot tempo di posizionamento del settore sotto la testina (latenza rotazionale)
  - Tlet tempo di lettura del settore (tempo di trasferimento)

Il tempo di seek

- Il tempo di seek è la somma tra il tempo iniziale per l'avviamento del braccio TI e il tempo che quest'ultimo vada dalla traccia attuale alla traccia desiderata
- Possiamo approssimare Tseek come
- $$Tseek = TI + K \times NT$$
- dove K è una costante che dipende dal disco rigido e NT è il numero di tracce da saltare

- Tseek è una variabile casuale discreta
- Il minimo è zero, il massimo dipende dal disco



Latenza rotazionale

- E’ una variabile casuale continua che dipende dalla velocità di rotazione del disco
- Il minimo è zero, il massimo è il tempo di una rotazione, la media è la metà

- Il tempo di una rotazione si ottiene come
- TR=60(secondi) / numero di rotazioni per minuto (rpm)

- Esempio: Disco da 7200 rpm
- Tempo di rotazione 60/7200 = 8,3 ms
- Latenza rotazionale tra 0 e 8,3 ms . Media 4,15 ms

Tempo di trasferimento

- IL TEMPO DI TRASFERIMENTO dipende dalla velocità di rotazione del disco. Il tempo è così calcolato:
- TR = Nb / r X NbT
- Ove, TR è uguale al tempo di trasferimento, Nb è il numero di byte da trasferire, è la velocità del disco in giri al secondo e NbT sono i numeri di byte su una traccia.
- IL TEMPO TOTALE MEDIO DI ACCESSO E' DATO DALLA SEGUENTE:
- TMA = TS + 1/2r + Nb/r X NbT
- Ove, TMA è il tempo totale medio di accesso, TS è il tempo medio di ricerca, 1/2r è il tempo medio rotazionale e Nb/r X NbT è il tempo medio di trasferimento.

Volatilità e persistenza

- Possiamo pensare allora che le stesse strutture dati in memoria centrale siano utilizzabili in memoria di massa
- Cambierebbero solo i modelli fisici, non quelli concettuali e logici
- Le prestazioni comunque ne risentirebbero, in funzione della minore velocità della memoria di massa

Strutture dati ed uso

- Può essere più utile invece capire il differente uso delle strutture dati in memoria centrale e su memoria di massa
- Le strutture dati in memoria centrale sono volatili e mantengono informazioni operative temporanee
- Le strutture dati in memoria di massa mantengono informazioni persistenti che permettono l'operatività dei sistemi organizzativi (aziende, istituzioni).
- Parliamo in questo caso di sistemi informativi o data bases

La Struttura Dati *Data Base*

- Può avere senso allora usare la struttura dati Data Base come meccanismo unificante il tema delle strutture dati su disco.
- Il suo modello logico può in questo caso essere ricondotto alle istruzioni del linguaggio SQL, standard ISO
- Le realizzazioni del suo modello fisico possono spiegare l'uso delle strutture dati su disco, in termini di utilità, pregi e difetti

Struttura dati Data Base: modello concettuale

- *Un Data Base è una collezione di dati operativi memorizzati su un supporto per computer, utilizzato dai programmi applicativi di una particolare azienda* (C. Date)

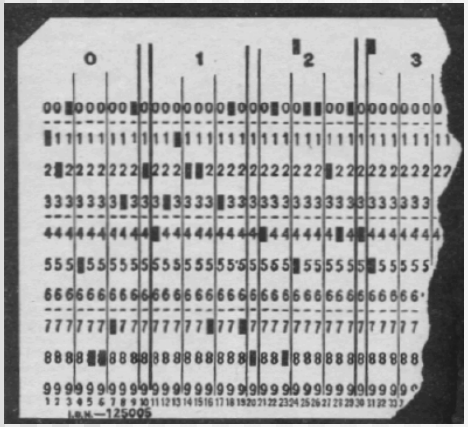
- Un Data Base contiene informazioni collegate tra di loro ed *informazioni sulle informazioni (meta-informazioni in un Dizionario dei dati)* in maniera da superare i difetti della gestione tradizionale degli archivi

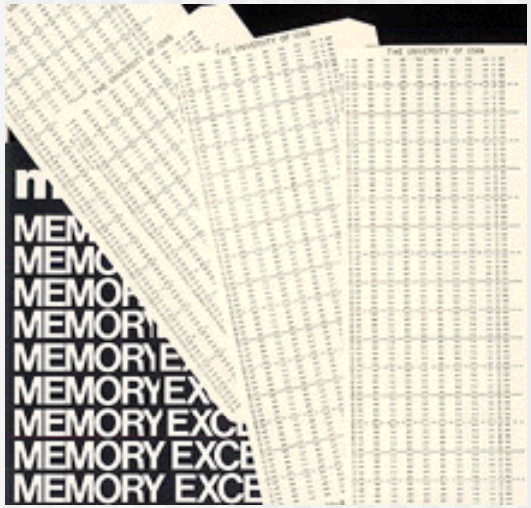
Struttura dati Data Base: modello logico

- Le sue proprietà corrispondono alle informazioni sulle informazioni (meta-informazioni in un Dizionario dei dati)
- I suoi metodi corrispondono alle istruzioni SQL
- Per capire invece i modelli fisici, sarà utile fare un po’ di storia

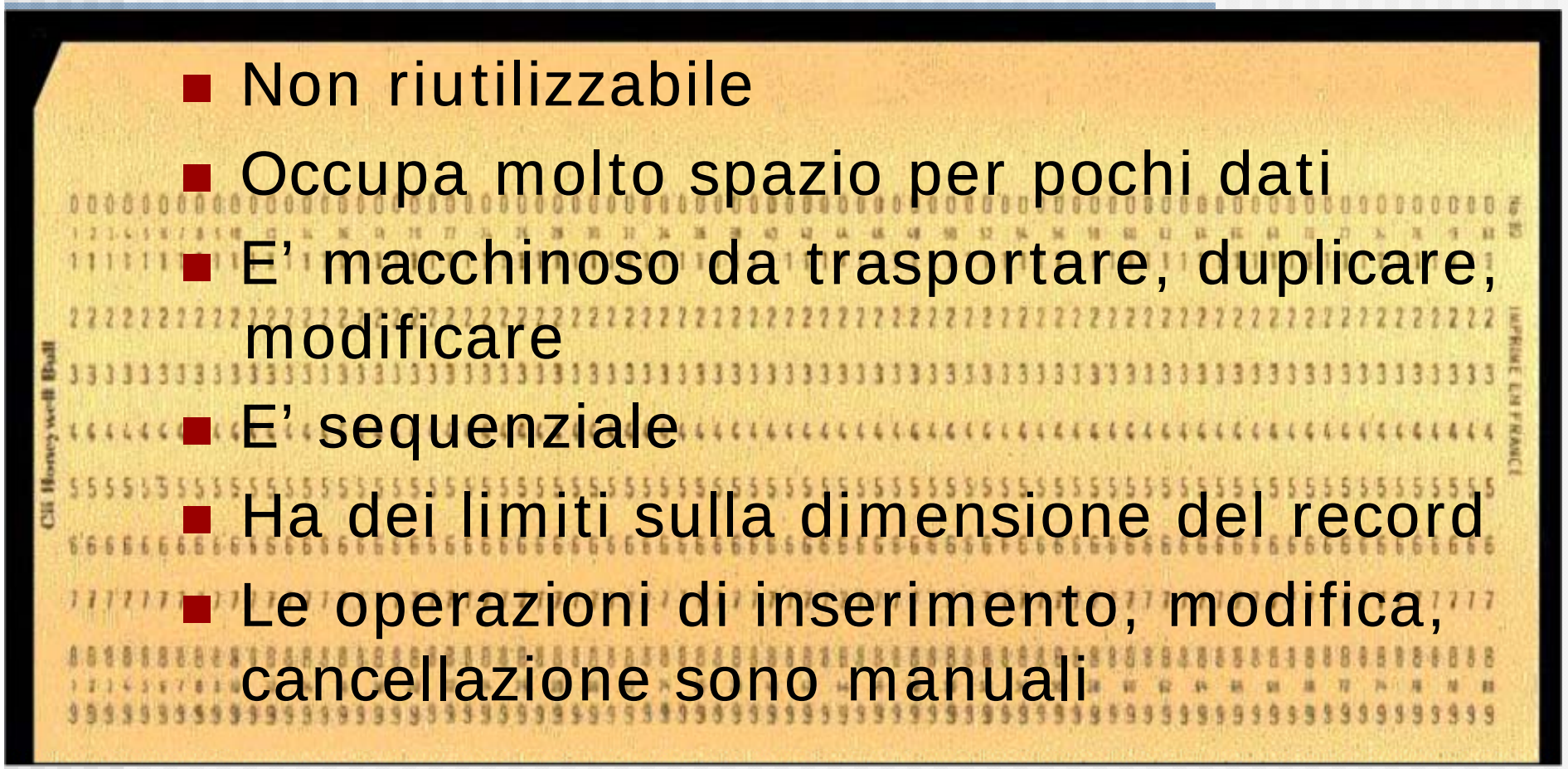
I dati tra anni ’60 e ’70

- Negli anni ‘50 i sistemi informativi su computer memorizzavano le informazioni su scheda perforata





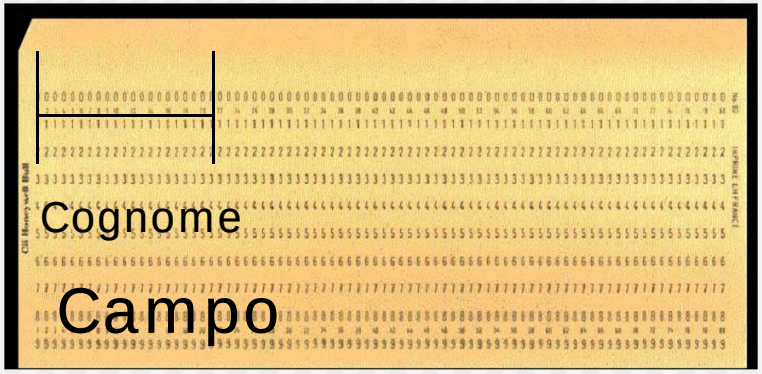
I Limiti di un Sistema Informativo su scheda



- Non riutilizzabile
- Occupa molto spazio per pochi dati
- E’ macchinoso da trasportare, duplicare, modificare
- E’ sequenziale
- Ha dei limiti sulla dimensione del record
- Le operazioni di inserimento, modifica, cancellazione sono manuali

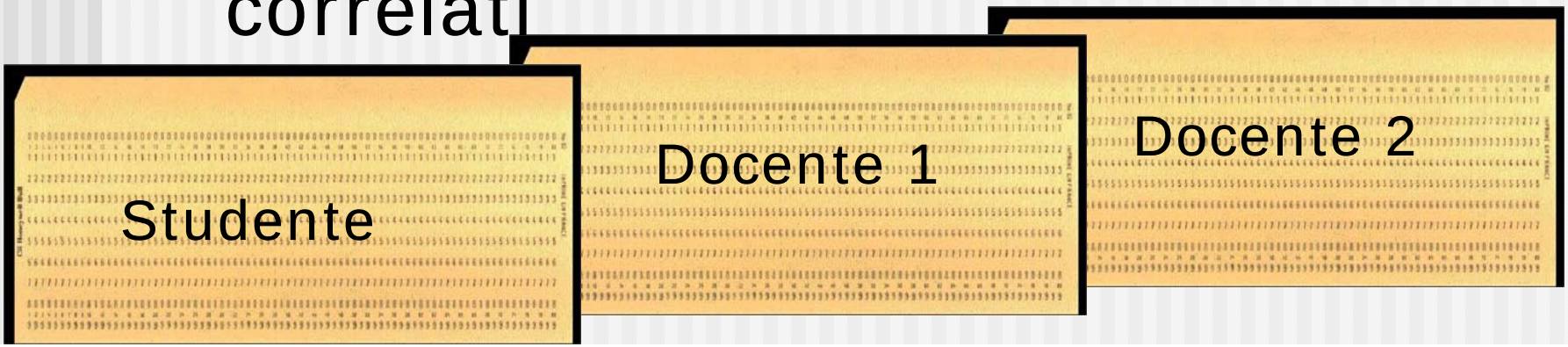
Vengono introdotti i concetti di base

- Record fisico, campo, record logico



Record fisico

- E’ possibile rappresentare archivi correlati



1953: le unità a nastro

- Riutilizzabile
- Occupa poco spazio
- E’ semplice da trasportare
- Contiene una gran quantità di dati
- Purtroppo, è sequenziale
- Le operazioni richiedono due unità a nastro

Figure 79. Tape Reel cartridge operation

1959: le unità a disco

- Ad accesso diretto
- Possibile il collegamento tra più archivi contemporaneamente
- Riutilizzabile
- Occupa poco spazio
- E’ semplice da trasportare
- Contiene una gran quantità di dati

La realizzazione di un programma che gestisce archivi su disco

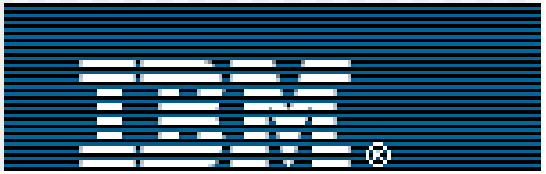
- Viene definito il “tipo record”
- Viene definito l’archivio (nome, supporto fisico, tipo record, organizzazione fisica)
- Vengono scritti i sottoprogrammi di inserimento, modifica, cancellazione, ricerca, stampa nel linguaggio di programmazione prescelto (COBOL, RPG, PL/I)

I difetti di una gestione tradizionale degli archivi

- Il modello dei dati è “cablato” nei programmi
- I controlli sono “cablati” nei programmi
- I programmi devono cambiare se cambia il modello logico o fisico dei dati
- Programmi diversi possono usare tipi record diversi per lo stesso archivio
- Ripetitività degli algoritmi
- I campi possono essere duplicati in più archivi
- Ridondanza, rischio di incongruenza e inconsistenza
- Mancanza di controllo degli accessi
- Difficoltà nell’accesso multiutente ai dati

1968: IMS, il primo DBMS IBM

- IBM progetta IMS come DBMS per il progetto APOLLO nel 1966
- Con Rockwell e Caterpillar, IBM lo utilizza per la prima volta nel 1969 per la gestione degli approvvigionamenti per la NASA
- Dopo 40 anni, è ancora in commercio
- Il suo creatore Vern Watts lavora ancora nel 2009 (anche se non ufficialmente) per IBM su IMS



Data Base: cos'è

- *Un Data Base è una collezione di dati operativi memorizzati su un supporto per computer, utilizzato dai programmi applicativi di una particolare azienda* (C. Date)

- Un Data Base contiene informazioni collegate tra di loro ed *informazioni sulle informazioni (meta-informazioni in un Dizionario dei dati)* in maniera da superare i difetti della gestione tradizionale degli archivi

Data Base: I pregi

- Il modello dei dati è nel Data Base
- I controlli di integrità sono nel Data Base
- Il controllo degli accessi è nel Data Base
- Raramente i programmi cambiano se cambia il modello logico o fisico dei dati
- Programmi diversi usano lo stesso modello per lo stesso archivio
- I programmi applicativi non contengono gli algoritmi di base ma usano un linguaggio di interrogazione e manipolazione per accedere ai dati
- L'esigenza di duplicare i campi viene ridotta
- Meno ridondanza, meno rischio di incongruenza
- Supporto nell'accesso multiutente ai dati

Gli Anni ’70: Vengono messi a punto gli algoritmi fondamentali

- Accesso Sequenziale
- Accesso Diretto Relativo
- Accesso Sequenziale con Indice
- Tecniche di Ricerca
- Ricerca Binaria
- Indice con B-Tree
- Tecniche *Hash*

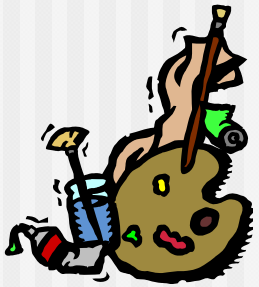
1971: Codd (IBM) mette a punto il modello relazionale

- Il modello relazionale dei dati guarda le informazioni come a tabelle con righe e colonne
- Le righe in qualche maniera rappresentano i records
- Le colonne in qualche maniera rappresentano i campi dei records
- Codd introduce anche un linguaggio per gestire i dati, SQL, oggi standard ISO

Assunzioni di Base

- I duplicati non sono permessi
- L’ordine non è importante
- I valori sono atomici
- Le colonne sono omogenee
- Le righe sono omogenee

L’importanza dei modelli dei dati



- Rappresentano la struttura delle informazioni
- Il modello concettuale dei dati rappresenta la struttura delle informazioni inerente al problema nel mondo reale
- Il modello logico dei dati rappresenta la struttura delle informazioni così come sono visibili all’utente di un DBMS
- Il modello fisico dei dati rappresenta la struttura delle informazioni così come risiede su memoria di massa

Costruttore e Distruttore

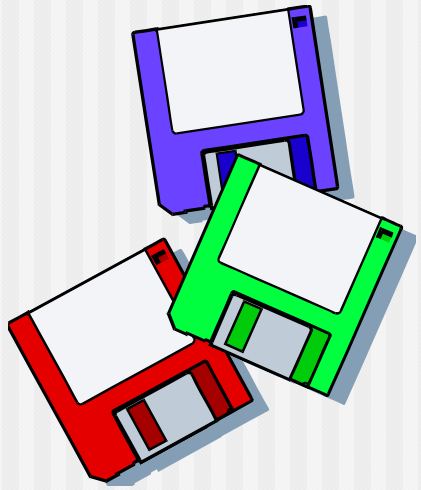
- Create Database nome
  - Crea una struttura dati database
- Drop Database nome
  - Distrugge una struttura dati database

Creazione di una tabella

- CREATE TABLE nome (definizione_colonna [, definizione_colonna ...]);
- Esempio:
- CREATE TABLE cliente (
  - code char(2) not null unique,
  - name char(15) not null,
  - st char(2) not null,
  - rating numeric(2)
-);

Tipi di campi

- NUMERIC[(lunghezza [,cifre_decimali])]
- DECIMAL[(lunghezza [,cifre_decimali])]
- DEC[(lunghezza [,cifre_decimali])]
- INTEGER
- INT
- SMALLINT
- DATE



Cancellazione di tabella

- DROP TABLE nome_tabella;
- Esempio
  - DROP TABLE clienti;

Inserimento in una tabella

- INSERT INTO nometabella
  - ([nomecolonna [,nomecolonna...]])
  - VALUES (valorecolonna [, valorecolonna]);
- Esempio:
- insert into clienti
  - (cust code, name, st, rating)
  - values('AA','Compugorp','WA',20);

Modifica di righe

- UPDATE nometabella
  - SET nomecolonna = valore
  - [, nomecolonna = valore ...]
  - WHERE condizionericerca ;
- Esempio:
- update cliente
  - set rating = rating + 5
  - where code = 'AA';

Cancellazione di righe

- DELETE FROM nometabella [WHERE espressionericerca]

- Esempio:
  - delete from cliente where rating < 10;

Ricerca in una tabella

- SELECT [DISTINCT]
  - nomecolonna [, nomecolonna ...]
  - FROM nometabella [, nometabella ...]
  - [WHERE espressionericerca]
  - [ORDER BY nomecolonna [, nomecolonna ...];
- Esempio:
  - select * from manu;
  - select code, mst from manu;