

Le basi del Data Base

Data Base Management System: sistemi per microcomputer

di Andrea de Prisco

Siamo giunti a una prima puntata conclusiva del nostro viaggio nei sistemi di gestione per basi di dati. Dopo aver parlato degli aspetti teorici dei modelli di dati generalmente usati, questo mese vedremo i Data Base dal punto di vista del lettore medio di MC: un utente di personal computer. Cosa è possibile e cosa è impossibile (o quasi) implementare su un microcomputer.

Settima parte

Micro, home o personal?

È difficile stabilire con esattezza a quale di queste tre categorie appartiene un qualsiasi computer. O perlomeno sembra impossibile delimitare un netto confine tra le tre fasce. In virtù anche dei continui cambiamenti di mercato e, perché no, di prezzo delle macchine.

Di macchine appartenenti alle tre categorie, è risaputo, ce n'è per tutte le tasche. Il Vic-20 è certamente un home, almeno a giudicare dal suo prezzo e/o dalle sue capacità.

L'M 20 della Olivetti è un personal, così come l'IBM Personal Computer.

Il Commodore 64, nato certamente come personal dalle capacità interessantissime (alla presentazione un paio d'anni fa, spacciato perfino come CP/M compatibile) sembra essere diventato il video game per eccellenza, a giudicare dall'età media dei suoi possessori e dal software disponibile, 90% giochi.

Che dire allora dei prezzi di questi mostriacattoli in continuo calo, in alcuni casi ribassati anche del 70%?

Una distinzione basata su valutazione a carattere prezzereccio, sembra essere dunque la meno attendibile. Ma allora, il nostro computer è un home, un personal o un micro?

E che dire delle periferiche? Esistono personal driver e home driver? Personal stampanti e home stampanti?

Qualcuno comincerà a chiedersi cosa c'entra tutto questo discorso con i data base, argomento di queste puntate. La chiave sta tutto nel sottotitolo: sistemi per micro computer.

Cioè?

Scopriremo che esistono Home Data Base, Personal Data Base e Micro Data Base.

Home Data Base

...o per meglio dire, Data Base D.O.C. (d'origine casereccia).

Secondo una recente definizione, una Base di Dati è un insieme di dati strutturati e permanenti, raggruppati in insiemi omogenei in relazione fra loro, organizzati con la minima ridondanza per essere usati da applicazioni diverse, in maniera controllata < Albano 84 >.

A tutt'oggi, il termine data base, di contro, è continuamente nominato (...e blasfemato) per indicare perfino semplici programmi Basic che permettono una mini-gestione di un mini-indirizzo su mini-videogiochi che tanto stanno facendo impazzire (a causa della nota game-dipendenza) ragazzetti di tutte le taglie.

Parlare di data base vuol dire fornire specifici strumenti per la definizione e il trattamento di grandi moli di dati, strumenti per assicurare l'integrità e la sicurezza degli stessi, nonché un vero e proprio linguaggio di programmazione usato per scrivere proprie procedure e utility che sfruttano la base di dati installata. Il produttore medio di software questo non lo vuole capire e spesso produce (secondo lui) data base che non sono degni nemmeno di essere chiamati sistemi di archiviazione.

L'opportunità di organizzare la base di dati in classi di elementi, e non in un unico insieme dalle dimensioni enormi, nasce essenzialmente da due necessità: minima ridondanza, duplicazione di informazione e massimo sfruttamento di quest'ultima. Anche se ne abbiamo già ampiamente parlato 5 numeri fa, per intenderci su questo punto, facciamo un esempio: vogliamo gestire i dati della segreteria studenti di una università, e in particolare gli studenti iscritti a un corso di laurea e gli esami dell'ordinamento dello stesso.

La situazione è mostrata in figura 1: abbiamo due classi, Studenti ed Esami, e l'associazione multipla tra questi. Multipla sta nel fatto (ovvio) che ogni studente può aver superato più di un esame e che ogni esame del corso di laurea è stato superato da più studenti. Avremo cioè che ogni elemento della classe Studenti sarà associato con uno o più elementi della classe Esami e viceversa.

Indipendentemente dal modello di dato usato (semantico, gerarchico, reticolare o relazionale) e a eventuali trucchetti, discusi nei numeri scorsi, per implementare tale situazione quando non direttamente possibile, la scomposizione in due classi di oggetti tra loro correlati permette ad esempio di conoscere (subito, facilmente) quali esami ha sostenuto un determinato studente, o quali studenti hanno superato un determinato esame, partendo (sempre ad esempio) dall'indirizzo di uno studente o dal nome di battesimo del professore che tiene un corso (e che presiede alla sessione d'esame).

Per di più, modificare la base di dati vista sopra, si riduce a informare il sistema che "Lo studente Tizio ha superato l'esame BlaBlaBla".

Personal Data Base

Non sono ancora Data Base effettivi, ma ci manca poco. A questa categoria appartengono potentissimi sistemi di archiviazione che permettono in alcuni casi perfino di programmare, con un opportuno linguaggio fornito per il trattamento dei dati, le applicazioni necessarie all'utente (generazione di rapporti, statistiche sui dati, ecc.).

L'unica vera limitazione sta nella mancanza del meccanismo delle classi che, come visto sopra, permette di modellare mol-

to facilmente aspetti assai comuni nella realtà.

In un sistema di archiviazione, per modellare la situazione studenti-esami si può procedere in due distinti modi: utilizzare un archivio unico o due archivi non (implicitamente) in relazione tra loro.

Dato che per ogni archivio che si definisce è possibile specificare il nome dei vari campi delle registrazioni, il primo modo consiste nell'inserire l'elenco degli esami sostenuti come proprietà del singolo Studente. Ogni elemento di questo super archivio, avrà come campi: Nome, Cognome, Matricola, Indirizzo, Telefono, DataDiNascita, Esame1, Esame2,... EsameN (se N sono gli esami del corso di laurea in questione).

Al momento dell'iscrizione un qualsiasi studente avrà tutti gli esami "vuoti". Man

abbia come numero di telefono 353637383 anche tutta quest'altra roba sarà presente cento volte nell'archivio.

È chiaro che tale situazione non salterà mai in mente a nessuno di implementarla così com'è. Meglio ricorrere a due archivi separati, uno contenente informazioni riguardanti gli studenti, l'altro gli esami di un corso di laurea. Si cerca cioè di spingersi maggiormente verso una specie (permetteteci questo termine) di DataBizzazione dell'archivio, purtroppo con notevoli limitazioni rispetto ai Data Base veri.

Inseriremo come proprietà degli studenti, oltre alle informazioni riguardanti nome, matricola, eccetera, lo spazio per i soli codici degli esami sostenuti (il codice identifica univocamente un esame).

L'archivio Esami avrà i seguenti campi: Codice, NomeEsame, NomeProfessore,

alla Biro-device (penna a sfera) si trasferiscono su supporto cartaceo (block-notes, Scottex casa o in mancanza d'altro anche il tavolo del computer) i codici degli esami da lui sostenuti. Prima di aprire l'archivio Esami si chiude l'archivio Studenti (non sempre è possibile mantenere aperti più archivi). Non resta che chiedere al sistema tutti gli esami che hanno come codice un codice trovato precedentemente presso lo studente Tizio e leggere direttamente su monitor i nomi cercati.

Se il sistema dispone di un proprio linguaggio di programmazione dotato di tutti gli operatori per il trattamento dei dati in memoria di massa, è possibile automatizzare questa procedura (chiamandola ad esempio EsaminatoriDi) rendendo il tutto ancora più vicino ai Data Base veri e propri.

Micro Data Base

A questo punto, se non avete a disposizione un IBM PC o un M 20, con tanto di disco rigido da 10 Mega in linea, scordatevi pure di veder realizzare queste cose sul vostro personal. Non siamo cattivi: è solo che per fare davvero le cose per bene, è ridicolo pensare a un Commodore 64 con tanto di 1541 appiccicato.

Comunque, per non lasciare a bocca asciutta l'utente medio, nel prossimo numero vedremo cosa sia possibile fare con un 64: presenteremo un data base che, seppur molto limitato in quanto a interrogazione e a quantità di dati trattabili, perlomeno avrà la possibilità di gestire i dati per mezzo di classi e (udite, udite!) con correlazione tra le registrazioni presenti in memoria di massa.

I Micro Data Base non sono altro che versioni leggermente semplificate dei Sistemi di gestione per Basi di Dati propri dei grossi calcolatori IBM, Honeywell, HP e consimili.

Le limitazioni riguardano essenzialmente il numero massimo di utenti che possono accedere al sistema contemporaneamente da più terminali (mediamente 8, ma in alcuni casi si può arrivare anche a 127) e dalla mole di dati trattabili, dovuta alla quantità di memoria (centrale e di massa) di cui si dispone.

Per quanto riguarda i sistemi monoutente (o personali se preferite) la tendenza principale è di fornire un prodotto quanto più facile da usare e con una flessibilità tale da permettere all'utente stesso di apportare delle modifiche all'organizzazione della Base di Dati adattandola alle sue mutevoli esigenze.

Per il resto, i Micro Data Base, sono strumenti potentissimi che permettono di trattare i dati allo stesso modo dei grossi calcolatori. Avremo MBD reticolari, relazionali con tutti gli operatori propri del modello, nonché potentissimi strumenti di programmazione per poter automatizzare le procedure che si desiderano.

Anche l'integrità dei dati e la sicurezza di questi è garantita dagli stessi meccanismi

LOG... contro il mal di black-out

Qui si parla di togliere e inserire elementi in una base di dati come fossero "bruscolotti". Non s'è detto, sebbene l'utente possa ignorarlo del tutto, che ogni registrazione in una base di dati non è... solo registrata.

Ad ogni inserimento bisogna aggiornare una marea di puntatori, avvertire gli elementi padre che è nato un altro figlio, tutti gli indici su chiavi secondarie devono essere aggiornati. Solo così sono possibili ricerche veloci al momento opportuno. Idem dicasi per l'operazione di cancellamento: bisogna fare macchina indietro in ogni direzione.

Non a caso molti sistemi di gestione di base di dati, dopo una richiesta di modifica e prima del nuovo Prompt rispondono "Commit" (trad. commesso, nel senso di commettere, lascia intendere qualcosa di molto scottante...).

E se manca la corrente nel bel mezzo di una modifica? Magari può anche verificarsi che l'elemento è stato tolto, ma se tutti i puntatori non sono aggiornati, la base di dati passa in uno stato inconsistente. Può darsi ad esempio che un elemento non sia più presente in una classe, ma risulta ancora essere figlio di un altro elemento in un'altra classe. O viceversa.

Dipende da quale operazione il sistema fa prima e quale dopo, sempre serialmente. In

ogni caso è un bel pasticcio, anche perché spesso non si può cancellare un elemento solo dagli indici o solo da una classe.

Per ovviare a questo genere di inconvenienti (già, non sempre la colpa è dell'Enel: potrebbe disintegrarsi una unità a dischi per colpa dei raggi cosmici...) è buona norma fare delle copie periodiche di tutta la base di dati su altro supporto, magari conservandolo in ambienti diversi dalla Sala Macchine.

Solo questo, è troppo banale: l'elemento essenziale è il giornale delle modifiche, o LOG, che è un apposito file contenente la lista di tutte le operazioni effettuate con successo dal momento dell'ultima copia della base di dati sino al giorno disgraziato.

Quando si esegue una modifica, questa viene prima inserita sul giornale, poi eseguita e se tutto è a posto, una apposita marca sempre sul LOG, indica che ha ottenuto successo e che quindi ha cambiato lo stato della base di dati (rimanendo però consistente).

Se avviene qualche inconveniente per il quale la base di dati passa in uno stato inconsistente, niente paura: si prende l'ultima copia della base di dati di cui si dispone e a partire da questa il sistema esegue una per una tutte le modifiche (con successo) riportate sul giornale. La base di dati si ritroverà nello stesso stato dell'ultima operazione eseguita con successo prima del Black-out: non resta che ripetere il comando di modifica, sperando che non succeda dell'altro.

mano che supererà esami si dovranno riempire le varie colonne della registrazione riguardante lo studente in considerazione.

È ovvio che se 100 studenti hanno superato l'esame: "Letteratura Italiana dal 1600 al 1700", tale stringa sarà necessariamente presente cento volte nell'archivio in cento posti diversi.

Se poi per ogni esame ci interessa anche il nome e il numero telefonico del professore che presiede alla sessione d'esame (il titolare del corso) la situazione, quanto a spreco di spazio, comincia a farsi preoccupante. Dovremo infatti inserire anche Nome... NomeN e Telefono1... TelefonoN come proprietà di ogni Studente. Posto che il professore dell'esame di sopra si chiami "Massimiliano Mazzanti Viendalmare" e

TelefonoProfessore. Per ogni studente che supererà un esame, bisognerà accedere all'archivio studenti e inserire in nuovo codice nella registrazione riguardante lo studente.

L'affare si complica un po' se vogliamo conoscere il nome di tutti i professori che hanno "esaminato" lo studente Tizio. Per prima cosa si apre l'archivio studenti (a mo' di semplici file su disco, generalmente ogni archivio prima di essere usato va aperto). Si accede alla registrazione con una semplice interrogazione al sistema: in quanto a queste nulla da recriminare: i sistemi di archiviazione sono particolarmente potenti in merito a selezione di elementi o ordinamenti di archivi secondo qualsiasi criterio.

Una volta selezionato lo studente grazie

dei sistemi più grossi, quali l'aggiornamento automatico del giornale delle modifiche e la possibilità di fare delle copie periodiche di tutta la base. Con questi meccanismi si può stare al sicuro praticamente da tutti gli incidenti compreso il Black-Out nel bel mezzo di una transazione (a causa di una modifica, la base di dati passa da uno stato a un altro). Nell'apposito riquadro a pag. 90 è mostrato come è realizzabile tale sicurezza.

Vedremo ora alcuni Data Base per microcomputer iniziando dal basso col Superbase 64 adatto all'ultradiffuso Commodore.

Superbase 64

Delle tre categorie sopra discusse, certamente questo appartiene alla categoria dei

personal Data Base, essendo, di fatto, più simile a un potentissimo sistema di archiviazione. Niente classi o correlazioni tra dati: la potenza del Superbase 64 è tutta nella facilità d'uso e nelle molteplici possibilità di interrogazione.

Questo "quasi" data base è di tipo a maschera: le registrazioni hanno un proprio formato di uscita su video, dichiarato assieme al tipo dei campi al momento della creazione di un file. Si ha a disposizione l'intero schermo per poter "disegnare" la forma della generica registrazione di ogni file creato. Per esempio, se si sta definendo un archivio indirizzi vogliamo il nome nella seconda linea dello schermo; immediatamente sotto il cognome; di seguito (o qualche linea più giù) vogliamo il campo indirizzo con a fianco la città e così via fino a completo riempimento di tutto lo schermo.

Anzi, se non dovesse bastare, è possibile richiederne altri per inserire ulteriori campi nella dichiarazione del nostro record.

Oltre a questo, nelle varie schermate possono essere aggiunti altri caratteri grafici per abbellire un po' il contesto.

Definita la maschera delle nostre registrazioni possiamo procedere all'inserimento dei dati. Il sistema visualizza la schermata col cursore lampeggiante nel primo campo. È possibile anche muoversi nei vari campi, semplicemente usando i tasti di controllo cursore. Per la ricerca di elementi si procede in modo analogo: il sistema visualizza la schermata con i vari campi vuoti, e dopo averne riempito qualcuno per permettere al computer di individuare la registrazione si preme [Return]. Ad esempio, vogliamo trovare la prima registrazione che ha come campo Città

I modelli di dato

Faremo un piccolo riassunto dei modelli di dato visti negli ultimi 4 numeri, con particolare riferimento al modo di concepire l'associazione tra elementi di classi diverse. Il primo modello analizzato è quello semantico: il più innovativo di tutti essendo anche l'ultimo nato (in senso buono).

Modellare particolari situazioni col modello semantico dei dati è abbastanza semplice: abbiamo ad esempio l'associazione Studenti-Esami di figura 1. Ogni studente ha superato più esami, ogni esame è stato superato da più studenti: associazione multipla. In linguaggio letterale, la registrazione riguardante uno Studente avrà un campo Nome, un campo Cognome, un campo Matricola, e una sequenza di Esami (quelli da lui superati). Un Esame avrà un campo Nome, un campo Codice e la sequenza di Studenti che l'hanno superato. Tradotto in un linguaggio di tipo semantico scriveremo qualcosa del tipo:

```
class Studenti <-> (Nome:string and Cognome:string and matricola: int and esami Superati: seq Esami)
```

mi Superati: seq Esami)

```
class Esami <-> (Nome:string and Codice:string and SuperatoDa: seq Studenti)
```

Il modello dei dati gerarchico, essendo il più anziano è anche quello più vincolato. Esiste un solo tipo di associazione, con diretta multipla e inversa univoca e ogni classe può essere al più codominio di un'altra classe. In figura 2 è mostrato un esempio rappresentabile col modello gerarchico. Un dipartimento ha più docenti e più studenti iscritti, ma tanto i primi quanto i secondi non possono appartenere a più di un dipartimento.

In figura 3 è mostrato un esempio reticolare. In questo modello ogni classe può essere codominio di più classi, ma il vincolo dell'unicità dell'inversa resta. Ogni Studente appartiene a un solo Dipartimento e ogni Dipartimento ha più Studenti iscritti. Idem dicasi per l'associazione Dipartimenti-Docenti. La terza associazione (univoca) tra Corsi e Docenti lega un corso al suo Docente.

I data base Relazionali devono il loro nome

al concetto di relazione (tabella di elementi, vedi fig. 4). Il meccanismo che permette associazione tra i dati è detto delle chiavi esterne, e si avvale di un particolare operatore relazionale detto JOIN. Il JOIN di due tabelle ne costruisce una terza, accoppiando a ogni elemento della prima tutti gli elementi della seconda che soddisfano una particolare condizione data assieme al comando. Per fare un esempio, immaginiamo che la relazione di figura 4 rappresenti nome e telefono dei clienti di una ditta. Supponiamo di avere un'altra tabella di tutti i clienti che non hanno ancora saldato un conto: avremo per ogni elemento un campo NomeCognome e un campo ImportoDovuto. Se vogliamo telefonare per un sollecito a tutti i clienti morosi faremo il join tra la tabella crediti e la tabella utenti con condizione di uguale NomeCognome. Otterremo in questo modo una tabella con elementi formati da tre campi: NomeCognome, ImportoDovuto e Telefono.

Dito indice alla mano, possiamo alzare la cornetta telefonica.

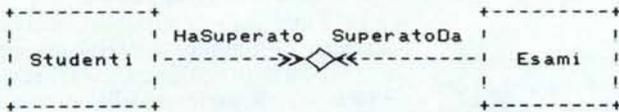


Figura 1

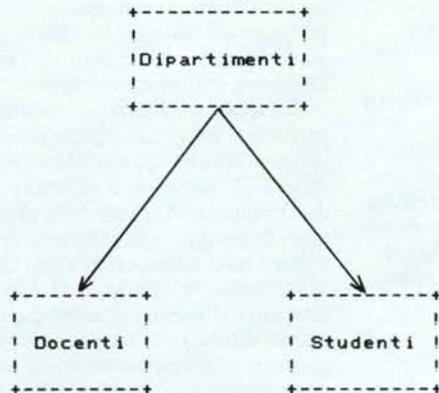


Figura 2

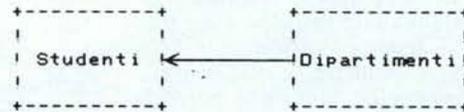


Figura 3

NomeCognome	Telefono
Mario Rossi	877877
Gino Verdi	765654
Luca Bianchi	546576
Nino Neri	868584

Figura 4

"Pisa". È sufficiente digitare Pisa nel corrispondente campo della schermata e battere [Return]. Il sistema si posizionerà sulla prima registrazione che soddisfa tale condizione (Città = Pisa). Da questa posizione del nostro file è possibile chiedere il successivo, il precedente o tornare in un sol colpo al primo elemento del file.

Per quanto riguarda i tipi di campi ammissibili, dobbiamo dire che anche qui la Precision Software (madre sia del Superbase 64 che del famosissimo Easy Script) non ha badato a spese. Sono infatti disponibili campi di tipo testo (lettere, numeri e simboli); numerici interi e decimali; campi di tipo Data (con calcolo automatico del corrispondente giorno della settimana) e di tipo Result ossia risultato di operazioni con altri campi, a mo' di tabellone elettronico. Per ogni registrazione è obbligatorio definire un campo chiave, tramite il quale è possibile un accesso molto veloce alla singola registrazione. Per finire la possibilità di scrivere propri programmi in un linguaggio simile al Basic, naturalmente fornito di tutti gli operatori per accedere alle registrazioni dell'archivio.

In parole tecniche (ne abbiamo parlato nel numero scorso) il Data Manipulation Language del Superbase 64 è ospite del Basic standard. È chiaro che quanto detto in queste righe è solo una lieve infarinatura delle varie possibilità offerte dal Superbase 64. Del resto non è questa la sede più opportuna per spiegarne appieno tutte le feature, sperando di poter presto ritornare sullo specifico tema con una prova su strada di questo particolare package.

dBASE II

Nato come sistema di gestione di base di dati per microcomputer dotati di sistema operativo CP/M, il dBASE II è stato successivamente adattato anche per l'uso sotto MS-DOS (microprocessori a 16 bit). Relazionale monoutente, questo sistema dispone di molte caratteristiche interessanti: si tratta certamente di un package destinato a un uso professionale.

Per iniziare diciamo subito che il dBASE II è un data base vero, e non (tanto per cambiare) un ennesimo sistema di archiviazione evoluto. Spiccano tra le sue caratteristiche principali un buon linguaggio di programmazione a se stante (dotato cioè di proprie strutture per il controllo del flusso), l'indipendenza fisica e logica tra organizzazione dei dati e programmi, e non ultima la possibilità di leggere archivi generati da altri linguaggi di programmazione quali il FORTRAN o il PASCAL.

Queste due ultime peculiarità fanno sì che eventuali (piccole) modifiche alla struttura dei dati non si ripercuotano sui programmi applicativi già scritti e che, passando (come spesso accade) da un sistema di archiviazione a un data base, è possibile leggere i vecchi file per caricare in una maniera più o meno automatica la base di dati che si vuole costruire.

Fra le limitazioni di questo data base (imposte essenzialmente dalle caratteristiche hardware delle macchine alle quali è destinato) annoveriamo la lentezza dei programmi applicativi, eseguiti caricandoli linea per linea dal disco e un non troppo flessibile data definition language: ogni registrazione può essere lunga al più un K e ogni campo del record al più 356 byte.

Essendo di tipo relazionale, la definizione della base di dati si ridurrà alla definizione delle varie tabelle (relazioni) da usare. Oltre a ciò, se interessa una maggiore velocità di ricerca per determinati campi di una tabella è possibile costruire automaticamente degli indici. Eventuali inserimenti o correzioni alla base di dati si ripercuotono implicitamente sulla ristrutturazione degli indici interessati.

Le operazioni possibili sui dati si riferiscono essenzialmente all'aggiunta di nuovi elementi in una tabella, al cancellamento (logico o fisico) di parte dei dati, e all'aggiornamento di record già esistenti. Oltre a queste è possibile una "totalizzazione" di alcuni campi ed eseguire l'operazione di giunzione (JOIN) tipica di ogni sistema relazionale.

La totalizzazione consiste nel fare la somma di alcuni campi numerici di tutti i record che hanno in comune altri campi. Facciamo un esempio, abbiamo la seguente tabella:

Articolo	Quantità
VIC 20	100
SINCLAIR SPECTRUM	120
CBM 64	90
APPLE 2/C	105
CBM 64	35
VIC 20	120
SINCLAIR SPECTRUM	120
VIC 20	100

riguardante le ordinazioni pervenute ad un distributore di personal computer. Totalizzando la tabella sui campi Articolo otterremo:

Articolo	Quantità
VIC 20	320
SINCLAIR SPECTRUM	240
CBM 64	125
APPLE 2/C	105

che corrisponde alla tabella degli articoli e dei totali delle quantità ordinate.

L'operazione di JOIN permette di creare tabelle a partire da due tabelle già esistenti: si usa per effettuare la famosa correlazione tra dati col meccanismo delle chiavi esterne, come già ampiamente discusso lo scorso mese e mostrato in sunto nel riquadro a pag. 91 di questo articolo.

MDBS

La sua sigla sta per Micro Data Base

System e, attualmente, si può considerare il più potente tra i data base per microcomputer. Seguendo il modello di dati reticolare, pur non attenendosi strettamente alle specifiche standard CODASYL, offre tutte le potenzialità dei sistemi più grossi, e per alcuni versi anche qualcosa in più. Ad esempio la possibilità di definire le associazioni multiple tra dati appartenenti a insiemi diversi.

Possiamo dividere l'MDBS in 5 moduli:

- 1) DDL (data definition language)
- 2) DMS (data management system)
- 3) DRS (dynamic restructuring system)
- 4) RTL (recovery/transaction logging system)
- 5) QRS (query system/report writer).

Il DOL serve per definire la base di dati che si vuole installare. Lo schema generale di una definizione è il seguente:

```
< nome della base di dati >
< drive da utilizzare >
< diritti di accesso >
< struttura dei vari insiemi >
< struttura delle associazioni >
```

I diritti di accesso servono per proteggere dati da accessi indiscreti. È possibile ad esempio definire una PASSWORD senza la quale nessuno è autorizzato a curiosare nella BD o solo a fare modifiche.

Per ogni insieme si definisce a sua volta la struttura interna (nome e tipo di ogni campo della registrazione) e per le associazioni, automatiche o manuali, si deve indicare l'insieme Padre, l'insieme Figlio e il tipo di connessione (biunivoca, univoca-multipia, multipla-univoca o multipla).

Il DML, col meccanismo delle sotto routine, permette di operare sulla base di dati. La struttura generale di un comando assume la seguente forma:

```
EO = CALL (Indirizzo, "ComandoVeroeProprio", ArgomentiLinguaggioOspite)
```

EO è una qualsiasi variabile che al ritorno dalla sottoroutine conterrà un valore relativo all'esito dell'operazione. Indirizzo è l'indirizzo di partenza della routine. Gli ArgomentiLinguaggioOspite servono per interfacciare il programma in esecuzione con DML.

Il modulo DRS permette di modificare dinamicamente la struttura della base di dati. È possibile ad esempio aggiungere nuovi insiemi, aggiungere campi a insiemi già esistenti, modificare associazioni, togliere insiemi. Tutto sempre senza dover reinserire dati in memoria.

Se nel corso di una operazione dovesse verificarsi un errore hardware (compresa l'improvvisa mancanza di corrente) il modulo RTL provvede a riportare la base di dati in uno stato consistente, grazie all'utilizzo di un giornale delle modifiche.

Per finire, tramite il modulo QRS è possibile accedere alla base di dati in modo interattivo, semplicemente digitando comandi di ricerca o di modifica, nonché generare rapporti o costruire nuovi insiemi con registrazioni che soddisfano particolari condizioni.

