

Introduzione a Oracle

Dott. Maurizio Boghetto

Indice

- 1 I Database Relazionali**
- 2 Il Modello Relazionale**
- 3 Structured Query Language**
- 4 Esempi di codice**
- 5 Appendici**

Capitolo primo

I Database Relazionali

1. Cos'è un database relazionale.

Microsoft Access, Oracle, Informix, Microsoft SqlServer sono sicuramente nomi che comunemente vengono associati alla gestione di archivi informatici. Ma cosa rappresentano tutte queste sigle e cosa hanno in comune ?

Tutti quanti sono RDBMS (Relational DataBase Management System) ovvero un insieme di software che si occupano di gestire le basi di dati ed in particolare le basi di dati Relazionali.

Tutti quanti, inoltre, comunicano per mezzo di un linguaggio detto SQL (Structured Query Language) che è il linguaggio standard delle basi di dati relazionali¹.

Ma cosa sono le Basi di Dati ?

E' difficile avere una concezione intuitiva delle Basi di Dati.

Al di là di quelle che possono essere le definizioni formali delle basi di dati, proviamo a fare questo ragionamento:

Un testo elettronico (per intenderci un documento di Microsoft Word) può essere intuitivamente considerato come un comune **un foglio a righe**.

Un foglio di calcolo (per intenderci un foglio Microsoft Excel) può essere intuitivamente pensato come un **foglio a quadretti** su cui fare dei calcoli, annotare dei dati e così via.

Ma il database sfugge a questa rappresentazione intuitiva.

Si può definire il database come:

UNA RAPPRESENTAZIONE DELLA REALTÀ.

E' certamente una realtà di nostro interesse, come può essere il magazzino di una impresa, la biblioteca di una scuola, la prenotazione di una stanza d'albergo.

Chi fa il database deve conoscere esattamente queste realtà che riprodurrà nella struttura da realizzare.

Non si tratta quindi di annotare i propri dati su di un foglio per catalogarli. Per questo ci sono i fogli di calcolo che su di una grande "tabellona" o su più tabellone li immagazzinano.

Si dovrà realizzare una struttura composta da un certo numero di tabelle collegate fra di loro da legami logici che siano l'espressione della realtà che vogliamo codificare.

Il modello attraverso il quale rappresentare questa realtà, e che si è affermato col tempo, è quello **Relazionale**. Per cui si stabiliscono dei legami tra le varie tabelle.

Ma perché usare un Database relazionale?

¹ Dopo aver definito cosa queste sigle hanno in comune occorre per correttezza anche dire le differenze. Mentre Access è un RDBMS client Oracle, SQLServer, Informix sono RDBMS server. Inoltre, ognuno di loro ha sviluppato un proprio dialetto del linguaggio SQL e la comunicazione avviene tramite dei "traduttori" del dialetto in SQL standard.

I motivi sono tanti, fra questi si segnala:

- **MIGLIORE GESTIONE DELL'INSERIMENTO, MODIFICA E CANCELLAZIONE DEI DATI.**
- **RIDUZIONE DELLA DUPLICAZIONE DEI DATI.**
- **POSSIBILITÀ DI INTERROGARE I DATI .**

Si pensi a quanto sia difficile gestire i propri dati inserendoli, modificandoli ed eliminandoli in un foglio di calcolo quando cominciano ad essere molti.

Realizzare una o più tabelle non correlate porta poi a duplicare i dati con un aumento sia dello spazio utilizzato sia del tempo richiesto per l'inserimento.

Un'altra caratteristica che rende il database (e quello relazionale in particolare) estremamente utile è la sua capacità di interrogare i dati sfruttando le relazioni esistenti fra essi al fine di ricavare informazioni utili a gestire la propria attività.

Occorre anche dire perché i database relazionali si sono affermati come struttura più diffusa nel mercato mondiale.

Le regole alla base dei database relazionali arrivano direttamente dalla teoria insiemistica. Per cui queste regole sono certe e trasparenti. Non sono strutture di basi di dati (gerarchico, ad oggetti ecc..) "proprietarie", ovvero diverse da produttore a produttore. Le fondamenta di tutti i database relazionali sono comuni e poggiano su basi scientifiche a tutti note.

Su questa base è stato realizzato un linguaggio, l'SQL appunto, che è divenuto uno standard per la creazione e soprattutto la interrogazione delle basi di dati relazionali. Investire nelle basi di dati relazionali permette quindi di possedere oltre che di una struttura solida anche di una struttura trasparente e non dipendente da un particolare produttore che può essere trasportata da una piattaforma all'altra con estrema facilità.

2. Il Modello dei Dati

Un modello di dati è uno strumento concettuale che consente al progettista di attribuire un certo significato (o interpretazione) ai dati e di manipolare i dati stessi. Si assegna una struttura ai dati attraverso appositi meccanismi di strutturazione previsti dal modello².

Il dato non dà di per se informazioni se non gli si fornisce una chiave d'interpretazione. Il modello di dati è uno strumento concettuale tramite il quale si può acquisire conoscenza da un insieme di dati altrimenti insignificante. L'organizzazione dei dati all'interno di questo schema attribuisce al dato una interpretazione che nel caso del modello relazionale viene definita come **rappresentazione della realtà di interesse** (Codd).

1. Indipendenza fisica e logica e i livelli di descrizione dei dati

Un requisito irrinunciabile dei DBMS è che devono garantire le proprietà d'*indipendenza fisica e logica*. Per *indipendenza (delle applicazioni dall'organizzazione) fisica (dei dati)*, s'intende la possibilità che i programmi applicativi non debbano essere modificati in seguito a modifiche dell'*organizzazione fisica dei dati*³. Le ragioni principali delle modifiche dell'organizzazione fisica dei dati possono essere :

- Un'evoluzione dell'uso dei dati può rendere necessario dover intervenire sull'organizzazione fisica della base dei dati.
- Cambiano i dispositivi fisici di memorizzazione, le tecniche di compattamento o di trasmissione dati, il sistema operativo oppure il suo sottosistema per la gestione degli archivi.
- Se la base di dati è distribuita su più nodi di una rete il DBMS può cambiare la locazione dei dati memorizzati oppure il tipo di calcolatore su cui sono memorizzati.

Per *indipendenza (delle applicazioni dall'organizzazione) logica (dei dati)*, s'intende la possibilità che i programmi applicativi non devono essere modificati in seguito a modifiche dello *schema logico*, per *aggiunte di nuove definizioni oppure per modifiche o eliminazione di definizioni già esistenti*⁴.

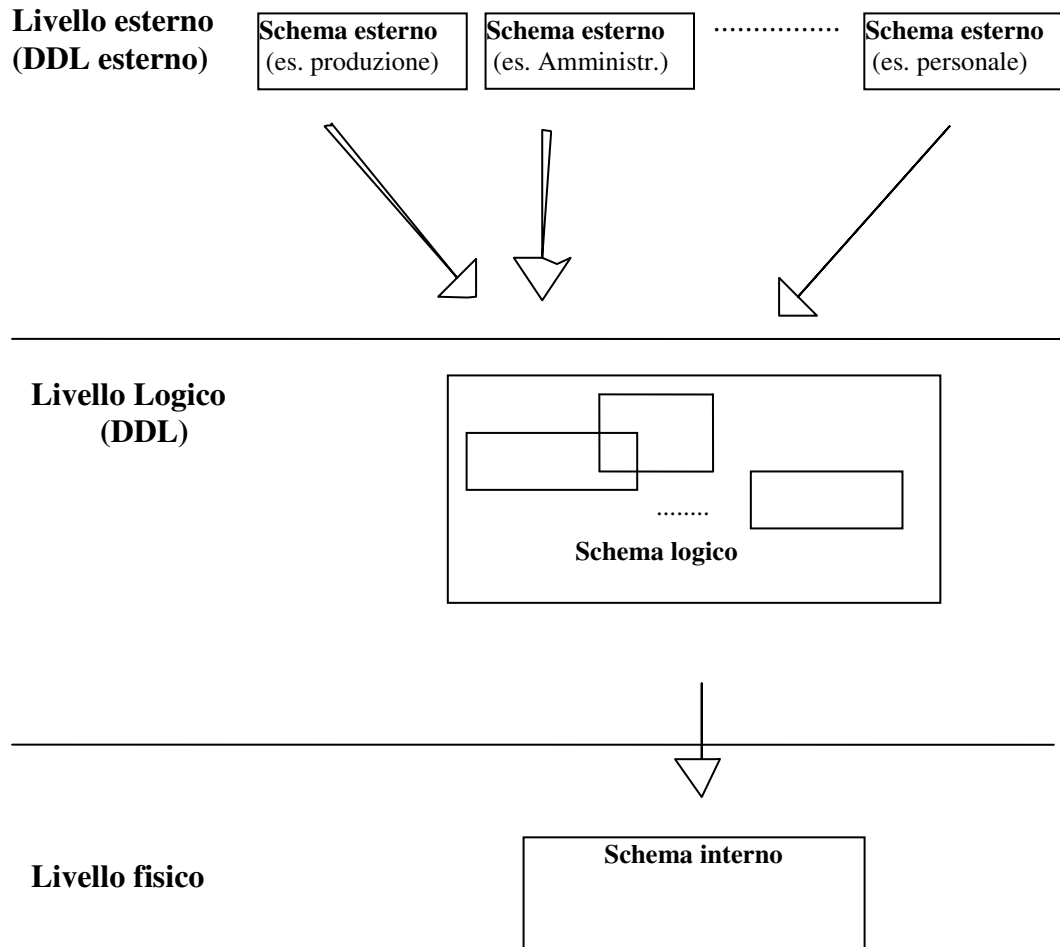
Per soddisfare questi requisiti, è stato proposto che i DBMS offrano tre livelli distinti di descrizione dei dati : *Schema logico, interno ed esterno*⁵.

² Corso di informatica generale - G. Callegaris CEDAM

Un'altra definizione è quella di Albano : "Un modello dei dati è un insieme di meccanismi di strutturazione, o di astrazione, per modellare una base di dati, con certi operatori e vincoli di integrità predefiniti"

³ Albano, Orsini- Basi di dati Boringhieri 1985

⁴ Albano, Orsini- Basi di dati Boringhieri 1985



Architettura a tre livelli di un DBMS

Lo schema logico descrive la struttura della base di dati non facendo nessun riferimento alla sua organizzazione fisica o al modo in cui vengono memorizzati i dati nelle memorie secondarie.

Lo schema interno è la descrizione di come sono organizzati fisicamente i dati nelle memorie di massa, e di quali strutture dati ausiliarie sono previste per facilitarne l'uso.

Lo schema esterno è una descrizione di come appare la struttura della base di dati ad una certa applicazione. Questo schema definisce quelle che si chiamano le viste, ovvero quelle porzioni della base di dati cui l'utente direttamente o tramite le applicazioni può accedere. In generale esistono più schemi esterni, uno per ogni gruppo omogeneo d'utenti. Essi però non sono indipendenti, poiché gli oggetti in comune hanno una rappresentazione unica nella base di dati e quindi le modifiche si riflettono su tutti gli utenti che ne fanno uso.

Fra questi livelli di descrizione dei dati devono esistere delle corrispondenze che vengono utilizzate dai DBMS per convertire le operazioni sugli oggetti virtuali accessibili da uno schema esterno in quelle sugli oggetti dello schema logico e, quindi, sui dati realmente presenti nel sistema, memorizzati secondo lo schema interno. Questi schemi sono gestiti dal DBMS e non dai programmi applicativi i quali, per accedere alla base di dati, comunicano al sistema con particolari accorgimenti a quale schema fanno riferimento.

Con questo sistema dei tre livelli di descrizione dei dati si garantisce sia l'indipendenza fisica, perché i programmi applicativi fanno riferimento allo schema esterno che non contiene informazioni sul modo in cui i dati sono organizzati fisicamente, sia l'indipendenza logica, perché i programmi fanno riferimento allo schema esterno e non allo schema logico.

⁵ Questo approccio è stato proposto dal comitato ANSI/X3/SPARC (1985).

2. Indirizzamento tramite indici

I gestori di database usano la tecnica degli indici per associare ad un dato presente nell'archivio un **puntatore** o **indice** che indica dove il dato **fisicamente** si trova. Lo scopo principale è quello di velocizzare l'accesso ai dati dell'archivio e quindi di velocizzare anche un'interrogazione dei dati (query) comportando però dei carichi aggiuntivi alla gestione dei dati. Infatti, quando vengono inseriti nuovi record il DBMS dovrà aggiornare anche gli archivi indice ad essi associati causando un incremento dei tempi d'esecuzione. Per questo motivo l'Amministratore di database dovrà valutare quanto incide l'operazione d'aggiornamento sulle applicazioni rispetto a quella di interrogazione. In questi casi si può far riferimento a due tipi di file : file dati e i file indice.

I file dati sono costituiti dalle informazioni della propria base di dati. Tipicamente, un file dati ha l'aspetto di una serie di record di lunghezza fissa composti da campi. Alcuni campi all'interno del record sono unici per quel record e specificano il percorso con il quale si accederà al record stesso, cioè si tratta solitamente delle chiavi primarie. Questi campi identificativi dei record vengono ripresi in un file separato denominato **file indice**. All'interno di un file indice le chiavi sono disposte secondo una di più strutture che permettono di accedere velocemente ai dati e di eseguire rapidamente inserimenti, cancellazioni e recuperi. Quasi tutti i sistemi di gestione delle basi di dati offrono più modi d'accesso fisico ai dati . I vari tipi d'operazione che si vogliono svolgere e la natura dei dati determinano quale modo o quali metodi diano le prestazioni migliori.

5. Progettare un Database (progettazione concettuale)

Per progettare un Database occorre:

Aver chiaro lo scopo per cui si vuole costruire la Base di Dati.

Conoscere la realtà di interesse.

E' logico che occorre trarre un effettivo beneficio dalla realizzazione di una base di dati per gestire efficacemente la propria realtà di interesse.

Per conoscere la realtà di interesse occorre realizzare tutta una serie di attività che costituiscono la progettazione vera e propria. In particolare occorre realizzare:

- **Una analisi conoscitiva della realtà di interesse**
- **Codificare questa realtà in uno schema (progettazione concettuale)**
- **Rappresentare la realtà secondo il modello relazionale (progettazione logica).**

Attraverso la prima attività si deve identificare le informazioni che servono e che si dovrà inserire nel DB. Queste informazioni, poi, andranno a comporsi in uno schema rappresentativo della realtà di interesse che sarà detto schema concettuale.

Questo schema viene rappresentato graficamente per mezzo del modello

Entità-Relazioni.

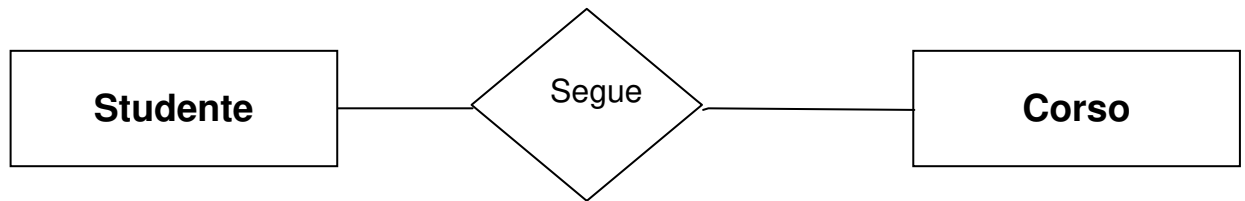
Gli elementi essenziali di questo modello sono:

- **Le entità:** L'entità è un "qualcosa" che nel contesto in esame ha una certa importanza. Ad esempio, se si vuol studiare una realtà scolastica, un'entità descriverà lo studente, del quale si potranno considerare le singole proprietà che interessano (detti attributi), come nome cognome, facoltà cui è iscritto, esami sostenuti e punteggio. Un'altra entità potrà essere il corso caratterizzato da un suo codice, descrizione numero ore annue ecc.

- **Gli attributi:** Con attributi si indicano dei dati che appartengono ad una entità.

- **Le Associazioni:** Indicano il tipo di relazioni fra le varie entità.

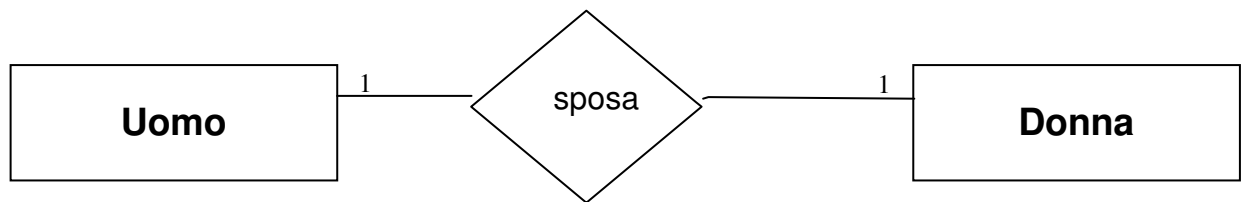
- **Le Relazioni:** Sono le relazioni fra le diverse entità. **Anche le relazioni possono avere i loro attributi**



Le relazioni possono essere di tre tipi:

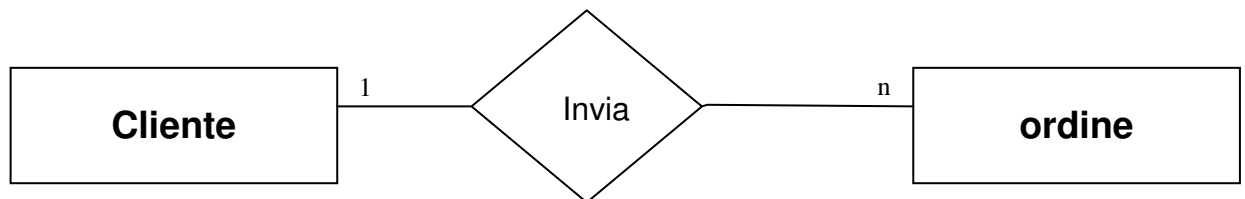
Associazioni 1 a 1

Si hanno quando ad un elemento della prima entità corrisponde uno ed un solo elemento della seconda.



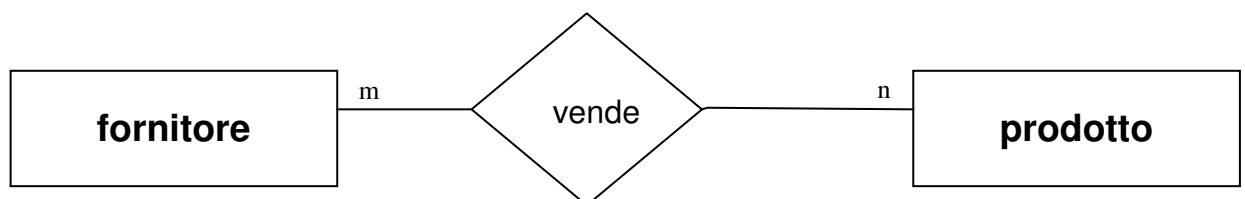
Associazioni 1 a n (oppure 1 ad infinito)

Si hanno quando ad uno ed un solo elemento di una entità corrisponde n elementi di un'altra entità.



Associazioni m a n (oppure infinito ad infinito)

Si hanno quando ad un elemento di una entità A corrispondono più elementi di una entità B e viceversa.



Come si vede le entità sono rappresentate graficamente da quadrati mentre le relazioni da rombi. A volte le relazioni sono rappresentate da delle semplici linee.

Una volta realizzato lo schema concettuale lo si codificherà secondo le specifiche del modello relazionale.

Attraverso lo schema ER si arriva a comporre una rappresentazione della realtà oggetto del nostro interesse. Questa realtà così codificata dovrà essere trasformata secondo le regole di un modello di rappresentazione dei dati (in questo caso il modello Relazionale).

I metodi adottati per realizzare una progettazione concettuale sono solitamente l'incontro di questi due:

1. **Top-Down.** Si parte da una situazione estremamente sintetica per poi passare a gradi di analiticità maggiori fino a realizzare uno schema adeguatamente rappresentativo della nostra realtà.
2. **Bottom-Up.** Si parte da una situazione estremamente analitica che viene passo passo sintetizzata fino a realizzare uno schema adeguatamente rappresentativo della nostra realtà.

Solitamente la progettazione concettuale rappresenta un processo in cui questi due metodi si incontrano.

Maggiore è la conoscenza della realtà di interesse, maggiore sarà il grado di analiticità con cui si inizia la nostra analisi.

Minore è il grado di conoscenza della realtà di interesse, maggiore sarà il grado di sinteticità con cui si inizia la nostra analisi.

Prodotto lo schema concettuale non rimane adesso che realizzare la cosiddetta progettazione logica, ovvero occorrerà trasformare lo schema secondo le regole relazionali.

Capitolo Secondo

Il modello relazionale.

1. Schema Logico

La fase della progettazione concettuale dei dati porta a definire lo Schema Concettuale.

Il modello concettuale cerca di dare una struttura ai dati evitando il più possibile i dettagli realizzativi, per cui con lo Schema Concettuale si otterrà un documento di base che descrive la struttura del sistema informatico in maniera astratta, vale a dire astratta dal modello concettuale prescelto. Quindi, è un documento che guiderà i progettisti, nella fase della progettazione logica, alla realizzazione dello Schema logico secondo un particolare modello scelto. Un modello dei dati può essere definito come *un insieme di concetti, descritti tramite un preciso formalismo, il cui scopo è quello di permettere una rappresentazione ed una manipolazione delle informazioni che costituiscono il mondo della realtà*⁶.

Il modello scelto per realizzare lo schema concettuale è quello relazionale elaborato da Codd nel 1970. Si tratta di un modello matematico per la descrizione dello schema logico in maniera indipendente dalla realizzazione fisica il quale trae le sue origini dalla teoria degli insiemi. Molti autori, in seguito, hanno approfondito l'argomento cercando di formalizzare i diversi problemi che l'utilizzo del modello dei dati ha suscitato, volendo rappresentare in un sistema informatico un frammento del mondo reale.

Alla base del modello relazionale si trova la relazione detta anche tabella o schema poiché questa è costituita su di una struttura di righe dette tuple e colonne dette domini della tabella.

Le testate della tabella A_i si dicono **attributi**, mentre il numero n dei domini viene detto **grado o arietà** della tabella. Le tabelle che hanno un solo attributo si dicono unarie, quelle con due attributi binarie, con tre ternarie e via dicendo. Il numero delle tuple si dice **cardinalità** ($m+1$).

Le proprietà delle relazioni possono essere così riassunte⁷ :

- *I valori di ogni colonna sono fra loro omogenei.* I valori di un attributo appartengono allo stesso dominio (interi, stringhe di caratteri, ecc..). Quindi, si può affermare che gli attributi rappresentano l'uso dei domini in una determinata tabella
- *l'ordinamento delle colonne è irrilevante.* Poiché sono sempre identificate per nome e non per posizione
- *l'ordinamento delle righe è irrilevante.* Poiché queste sono identificate per contenuto e non per posizione
- *Lo schema di una tabella $R(A)$ è un'espressione che contiene il nome della tabella seguito dall'elenco di tutti gli attributi della tabella stessa. Può essere sia l'intestazione della tabella sia questa espressione :* Prodotto (codice, descrizione, giacenza)

⁶ Schiavetti :Database, Jackson 1985

⁷ Atzeni, Batini, De Antonellis :Introduzione alla teoria relazionale, Masson

2. Chiavi ed attributi di una relazione

Il concetto di chiave viene elaborato nella teoria relazionale poiché le tabelle devono essere costruite in modo tale che ogni tupla deve essere distinta per mezzo di uno o più attributi. Quindi, scopo delle chiavi è identificare univocamente queste tuple. Si distingue fra superchiave, chiave candidate e chiave primaria. La **superchiave** è l'insieme degli attributi che identificano univocamente una tuple. La **chiave candidata** è in numero minimo di attributi necessari per identificare univocamente una tuple. Può essere definita come *una superchiave dalla quale possono essere eliminati attributi senza distruggere la proprietà della identificazione univoca della tuple. Per cui una chiave candidata è una superchiave con la proprietà di non ridondanza*⁸.

Fra le chiavi candidate, che possono essere più di una, se ne sceglierà una che ha il minor numero di attributi e che, a parità di attributi, ha il minor numero di caratteri. Questa è la **chiave primaria** e sarà la chiave scelta fra le chiavi candidate a rappresentare univocamente una tuple.

Rispetto alla chiave della tabella si possono distinguere gli **attributi primi**, cioè quelli che fanno parte di almeno una chiave candidata, e i restanti **attributi non primi**. Infine, quegli attributi che possono costituire la chiave di altre tabelle sono detti **chiave esterna** e permettono di stabilire associazioni fra le tabelle che rappresentano diverse entità.

Però dopo tutto questo parlare è meglio che vi faccia vedere una tabella.

Cod lettore	Cognome	Nome	Indirizzo	Città	Recapito telefonico
001	Paolino	Paperino	Via di Paperinia 51	Paperopoli	0034/45789
002	Il Gallico	Asterix	Via Del Menir 64	S.t Malo	091/567989
003	Dei Paperoni	Paperone	Via Del Deposito 1	Paperopoli	5689/487451
004	Bonaparte	Napoleone	Via dei Superbi 1	Porto Azzurro	0587/984512
005	Benso	Camillo	Via Monginevro 487	Torino	011/784645
006	Il Druido	Panoramix	Via del Menir 66	S.t Malo	091/4566654
007	Bond	James	Via Degli intrepidi 7	Frascati	007/007007

Riga

Campo

Colonna

Notate una cosa:

Se interessa sapere la città dove abita Paperon dei Paperoni basta trovare la riga che lo riguarda ed identificare la colonna con attributo Città. Così, scorrendo la colonna città si può identificare tutti i lettori che hanno preso in prestito un libro che abitano in una certa città.

Questo è in estrema sintesi il funzionamento della ricerca dei dati all'interno di un DB Relazionale: le ricerche vengono fatte per colonne (la colonna Città) e per riga (Paperon dei Paperoni)

3. Traduzione dei diagrammi E|R di uno schema concettuale in un modello relazionale

Si può tradurre abbastanza agevolmente gli schemi E|R secondo le strutture delle basi di dati relazionali seguendo, però, alcune regole.

⁸ Nazzini, Sanges, Vaccaro :Introduzione ai Data Base relazionali