

CAP6

PROGETTAZIONE

DI UNA BASE DI

DATI

Ing. Daniele Corti



Copyright © Ing. Daniele Corti 2013

www.ingdanielecorti.it

Tutti i diritti sono riservati a norma di legge e a norma delle convenzioni internazionali.

Ver.1.0

PREREQUISITI

- ✓ Riconoscere gli archivi di dati e i database.

OBIETTIVI

- ✓ Saper descrivere le fasi di progettazione di un database.

ARGOMENTI

- ✓ Introduzione.
- ✓ Definizione di Base di Dati.
- ✓ Definizione di DBMS.
- ✓ Ciclo di vita del Sistema Informativo.
- ✓ Fasi della progettazione di un DB: concettuale, logica e fisica.
- ✓ Esempio.

CAP 6 - PROGETTAZIONE DI UNA BASE DI DATI

INTRODUZIONE

Nello svolgimento delle attività, sia a livello individuale che nelle organizzazioni è basilare avere un sistema informativo per la registrazione, consultazione, organizzazione e gestione efficiente delle informazioni. Il sistema informativo risponde alle esigenze dell'organizzazione stessa.

La progettazione (design), in generale, può essere considerata essenzialmente un processo di **analisi** del contesto da rappresentare, e di astrazione attraverso l'utilizzo di un **modello** che descrive, mediante uno schema, le sole informazioni più significative scelte in base al **contesto** stesso.

Progettare una Base di Dati significa definire struttura, caratteristiche e contenuto: si tratta di un processo nel quale occorre prendere molte decisioni strategiche attraverso l'utilizzo di metodologie: strategie e metodologie sono fondamentali per la realizzazione di un prodotto di alta qualità.

La progettazione, in definitiva, è l'attività svolta dal progettista per la costruzione (realizzazione) di qualsiasi oggetto (semplice o complesso, fisico o astratto) / prodotto attraverso la stesura di un progetto.



In questo modulo si vedrà come sia possibile progettare una Base Dati Relazionale attraverso le fasi di progettazione concettuale, logica e fisica, che permette, partendo dai requisiti dell'utente, di arrivare a produrre strutture di DataBase di buona qualità.

DEFINIZIONE DI BASE DI DATI

Una base dati (Data Base) è un insieme di dati correlati, organizzati in modo che possano essere manipolati e che possano evolvere nel tempo.

Un database deve soddisfare i seguenti requisiti:

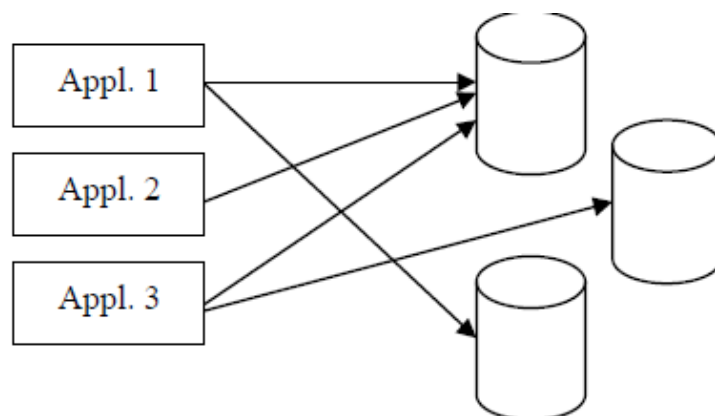
- i dati devono essere organizzati con ridondanza minima, ossia non devono essere inutilmente duplicati per evitare spreco di risorse di memorizzazione e, soprattutto, per evitare l'onere della gestione di copie multiple che possono mettere a rischio la consistenza e l'affidabilità dei dati;
- i dati devono essere utilizzabili contemporaneamente da più utenti, evitando che ognuno crei una copia propria degli stessi; deve esistere un'unica versione dei dati a cui gli utenti accedono secondo specifici diritti. Inoltre sono necessarie delle tecniche che consentano di evitare che l'attività dei vari utenti generi conflitti per l'uso contemporaneo degli stessi dati.

I Data Base più utilizzati sono quelli relazionali basati su strutture tabellari fra loro correlate.

DEFINIZIONE DI DBMS

I DMBS (Database Managment System) sono strumenti software che gestiscono in maniera efficiente ed efficace le informazioni contenute in una Base Dati.

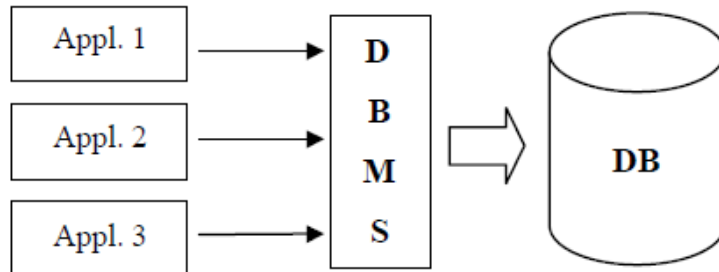
Come abbiamo già visto, prima dello sviluppo dei DBMS l'approccio che veniva applicato al problema dell'archiviazione prevedeva l'uso diretto delle strutture del File System:



Utilizzando questa strategia, le applicazioni accedono direttamente agli archivi, quindi ognuna deve conoscere la struttura interna degli archivi e le relazioni tra i dati e deve evitare la duplicazione degli stessi. Inoltre la non volatilità dei dati e la gestione degli accessi contemporanei di più

applicazioni agli archivi viene relegata a strati software non specializzati per tali compiti, quali il Sistema Operativo.

I sistemi che gestiscono i DB (DataBase) invece hanno un componente specializzato a svolgere tale compito:



La precedente figura mostra come le applicazioni rivolgono al DBMS le proprie richieste di accesso alla base di dati, il quale gestisce i dati svincolando le applicazioni da tale onere. Quindi il DBMS è un modulo, specializzato nella gestione del DB; a cui tutte le applicazioni si rivolgono per accedere ai dati. Si ottiene così un triplice scopo: da una parte le funzionalità di gestione del database sono raggruppate in un unico insieme, dall'altra le applicazioni risultano alleggerite e quindi più veloci da realizzare e, soprattutto, nessuna potrà effettuare operazioni scorrette sul database.

CICLO DI VITA DEL SISTEMA INFORMATIVO

Il progetto di una base di dati si inserisce nel ciclo di vita del sistema informativo (insieme di informazioni utilizzate, prodotte e trasformate da un'azienda durante l'esecuzione dei processi aziendali; non va confuso con il sistema informatico, che indica la porzione di sistema informatico che fa uso di tecnologie informatiche e automazione; il sistema informatico è quindi è un insieme di computer, composti da hardware e software, che elaborano dati e informazioni per restituire altri dati e informazioni utili) comprendente in genere le seguenti attività:

1. Studio di fattibilità.
2. Raccolta e analisi dei requisiti.
3. Progettazione.
4. Implementazione.
5. Validazione e collaudo.
6. Funzionamento.

Studio di fattibilità

Serve a definire, in maniera per quanto possibile precisa, i costi delle varie alternative possibili e a stabilire le priorità di realizzazione delle varie componenti del sistema.

Raccolta e analisi dei requisiti

Consiste nella individuazione e nello studio delle proprietà e delle funzionalità che il sistema informativo dovrà avere. Questa fase richiede una interazione con gli utenti del sistema e produce una descrizione completa, ma generalmente informale, dei dati coinvolti (anche in termini di previsione sulla loro frequenza). Vengono inoltre stabiliti i requisiti software e hardware del sistema informativo.

Progettazione

Si divide generalmente in progettazione dei dati e progettazione delle applicazioni. Nella prima si individua la struttura e l'organizzazione che i dati dovranno avere, nell'altra si definiscono le caratteristiche dei programmi applicativi. Le due attività sono complementari e possono procedere in parallelo o in cascata. Le descrizioni dei dati e dei programmi prodotte in questa fase sono formali e fanno riferimento a specifici modelli.

Implementazione

In questa fase si decide quali componenti HW/SW si devono usare per popolare la base dati.

Consiste nella realizzazione del sistema informativo secondo la struttura e le caratteristiche definite nella fase di progettazione. Viene costruita e popolata la base di dati e viene sviluppato il codice dei programmi.

Validazione e collaudo

Verifica del corretto funzionamento della base dati e controllo dati.

Serve a verificare il corretto funzionamento e la qualità del sistema informativo. La sperimentazione deve prevedere, per quanto possibile, tutte le condizioni operative.

Funzionamento

In questa fase il sistema informativo diventa operativo e richiede, a meno di malfunzionamenti o revisioni delle funzionalità del sistema, solo operazioni di gestione e manutenzione.

Occorre osservare che oltre alle attività sopra citate, viene oggi spesso effettuata anche una attività detta di **prototipizzazione**, che consiste nell'uso di specifici strumenti software per la realizzazione

rapida di una versione semplificata del sistema informativo, con la quale sperimentare le sue funzionalità. La verifica del prototipo può portare a una modifica dei requisiti e una eventuale revisione del progetto.

Poiché i dati hanno un ruolo centrale nei sistemi informativi si giustifica uno studio autonomo relativo alla progettazione delle basi di dati che si individua nella terza fase del ciclo di vita.

FASI DELLA PROGETTAZIONE DI UN DB

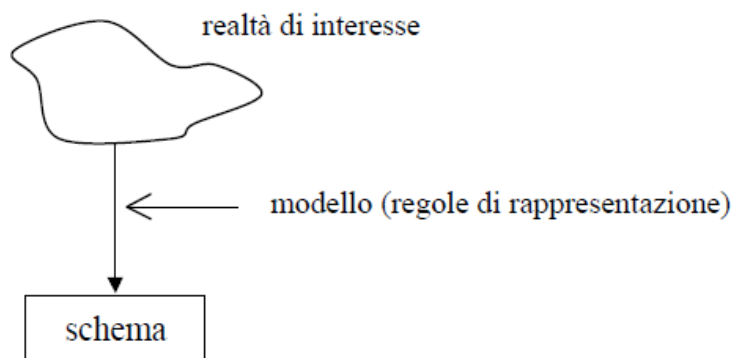
I passi fondamentali da compiere per progettare un Data Base relazionale si articolano in tre fasi principali da effettuare in cascata:

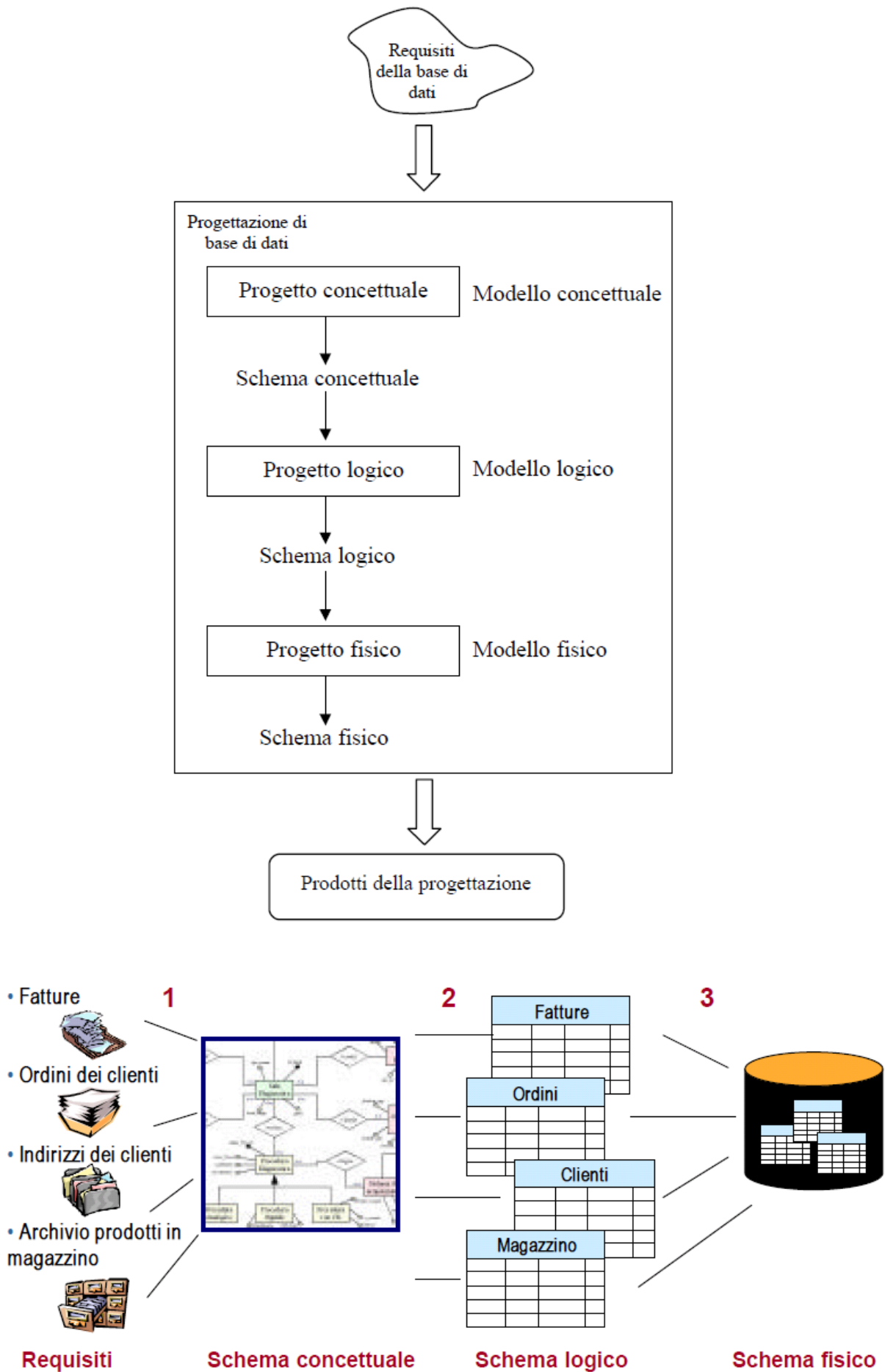
1. PROGETTAZIONE CONCETTUALE.
2. PROGETTAZIONE LOGICA.
3. PROGETTAZIONE FISICA.

Questa progettazione si basa sul concetto di tenere separate le decisioni relative a “**cosa**” rappresentare in una Base Dati (prima fase), dal quelle relative a “**come**” farlo (fasi successive).

Ogni fase si riferisce a un livello di **astrazione** nella rappresentazione dei dati e delle relazioni tra essi, e ha lo scopo di separare le attività di risoluzione dei problemi e di garantire la possibilità di modificare delle soluzioni adottate ai livelli inferiori senza dover riprogettare quanto definito nei livelli superiori.

A ciascuna fase di progettazione corrispondono diversi modelli per la rappresentazione dei dati, ovvero tecniche per la rappresentazione degli aspetti rilevanti della realtà da modellare, definite da strumenti e vincoli specifici. La rappresentazione generata seguendo le regole del modello viene detta **schema**.



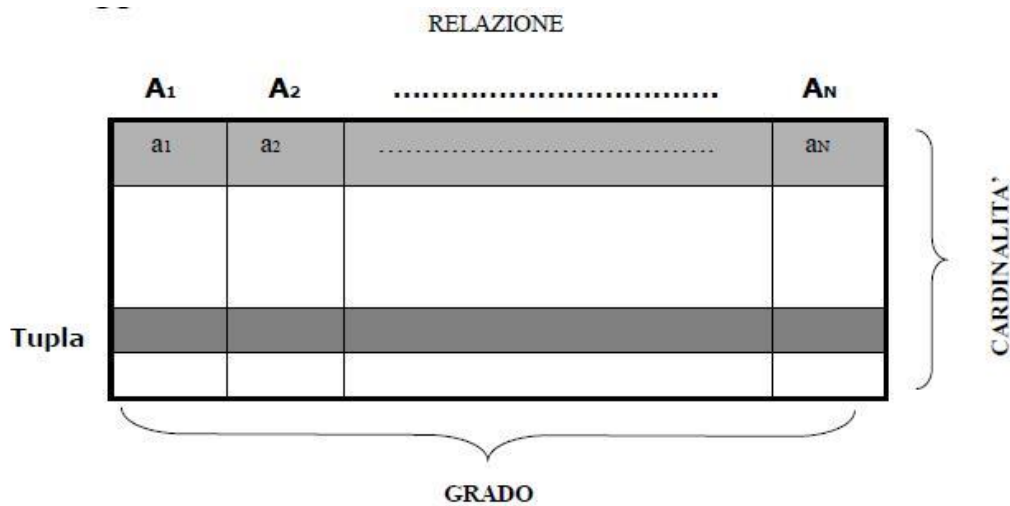


FASI DELLA PROGETTAZIONE DI UNA BASE DI DATI (DB)		
PROGETTAZIONE	SCOPO	MODELLO PIU' UTILIZZATO
CONCETTUALE	Costruire e definire una rappresentazione (astratta) FORMALE , corretta e completa della realtà di interesse (che è informale) indipendentemente dal tipo di DBMS utilizzato. È una rappresentazione SEMPLIFICATA che dovrà comunque contenere tutti e soli gli aspetti interessanti per la gestione dell'azienda. È il livello più vicino all'essere umano e più lontano dal computer.	ER: Entità - Associazione
LOGICA	Trasformare il modello concettuale in un modello logico dipendente strettamente dal tipo di software DBMS utilizzato. Il modello logico è una rappresentazione dei dati basata sull' AMBIENTE database scelto per la programmazione. In questa fase viene adottata anche la tecnica di normalizzazione al fine di rendere più efficiente il DB (ottimizzazione).	Relazionale: il DB viene descritto dalle relazioni (tabelle)
FISICA	Implementare lo schema logico definendo tutti gli aspetti fisici di memorizzazione e rappresentazione in memoria di massa. Si stabilisce come le strutture a livello logico debbano essere organizzate negli ARCHIVI e nelle strutture del FILE SYSTEM ; esso dipende quindi non solo dal tipo di DBMS utilizzato, ma anche dal SISTEMA OPERATIVO e in ultima istanza dalla PIATTAFORMA HARDWARE del sistema che ospita il DBMS . È il livello più vicino al computer e più lontano dall'essere umano.	Software DBMS che gira un SO

Nel modello ER (Entity/Relationship, Entità/Associazione) gli oggetti astratti o concreti della realtà d'interesse vengono rappresentati attraverso il concetto astratto di **entità**; ogni entità possiede degli **attributi** (che sono le caratteristiche essenziali dell'oggetto che si sta studiando); ogni entità può essere messo in correlazione logica con altre entità al fine di stabilire delle **associazioni** fra le entità stesse.

Traducendo il modello ER nel modello Relazione le entità prendono forma ovvero diventano delle **tabelle** (più correttamente delle **relazioni**, concetto puramente matematico). Questo è il modello più semplice ed efficace, fra quelli logici, perché è più vicino al modo consueto di pensare a come archiviare i dati.

Questo modello ER fu proposto da Edward F. Codd, ricercatore IBM, nel 1970.



Termine	Descrizione
Grado	Numero delle colonne della tabella, numero attributi
Cardinalità	Numero delle tuple
Tupla	Riga di una tabella (Record)
Relazione	Insieme di tuple
Dominio	Insieme dei valori assunti da un attributo
Tabella	Rappresentazione di una relazione

MODELLO CONCETTUALE

Il modello Concettuale, introduce una rappresentazione grafica dello schema concettuale. Esso prevede come prima cosa l'individuazione delle entità, ossia degli oggetti concreti o astratti, rilevanti per il sistema informativo.

Graficamente i tipi di entità sono rappresentati con il loro nome racchiuso dentro un rettangolo (vedi figura in basso). Le proprietà caratteristiche di ciascun insieme di entità vengono descritte mediante l'uso di attributi, che si possono distinguere in 3 categorie:

ATTRIBUTI SEMPLICI: sono quelli che hanno un tipo semplice (integer, real, char).

ATTRIBUTI COMPOSTI: sono quelli di tipo record, cioè in cui esistono dei sotto attributi.

ATTRIBUTI MULTIPLI: sono quelli di tipo sequenza, cioè in cui esiste un numero variabile di sotto attributi.

Le dipendenze o associazioni (dette anche relazioni) di interesse informativo tra i dati da rappresentare vengono espresse nello schema entità - associazione, mediante relazioni tra le corrispondenti entità. Graficamente un insieme di associazioni viene rappresentato da un arco che collega i rettangoli (o entità), che raffigurano i due insiemi di entità correlati, interrotto da un rombo che contiene il nome dell'insieme di associazioni.

Un'associazione viene classificata in base alla sua forma, e può essere del tipo:

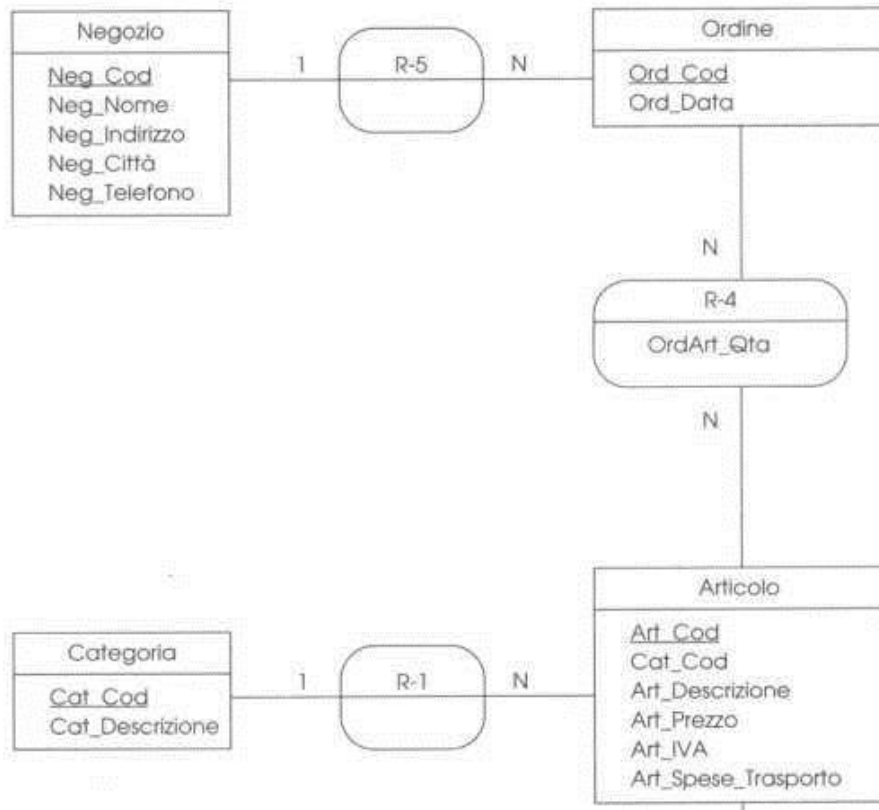
1 : 1 quando ad un elemento del primo insieme corrisponde uno ed un solo elemento del secondo insieme, e viceversa.

1 : N quando ad ogni elemento del primo insieme corrispondono più elementi del secondo, mentre ad ogni elemento del secondo corrisponde un elemento del primo.

N : N quando ad ogni elemento del primo insieme possono corrispondere più elementi del secondo insieme, e viceversa.

ESEMPIO

Schema Concettuale di Data Base



SCHEMA LOGICO

Un modello (o schema) logico di dati è una tecnica di organizzazione e di accesso ai dati, che deve essere utilizzata dal DBMS. A differenza dello schema concettuale, quello logico dipende strettamente dal modello logico di rappresentazione dei dati utilizzato dal DBMS (gerarchico, reticolare o relazionale). Si farà riferimento in particolare allo schema logico utilizzato nei DBMS relazionali, denominato schema logico relazionale.

In questa fase si parte dallo schema E-R ottenuto nel livello precedente, per ottenere un altro schema, detto anche schema logico. Questo schema spiega concretamente come realizzare il vero e proprio database, cioè gli archivi e i *collegamenti* tra gli archivi. A seconda di come si preferisca realizzare questi collegamenti, si può adottare un modello a scelta tra i seguenti: modello

relazionale, modello gerarchico, modello reticolare e modello ad oggetti. Studieremo più avanti questi modelli, in particolare il modello logico relazionale e come esso realizza i collegamenti tra gli archivi.

Lo Schema Logico Relazionale ha una struttura simile al diagramma entità, differisce da esso per l'esplicita dichiarazione delle chiavi primarie, l'inserimento delle chiavi esterne e l'aggiunta di opportune tabelle per rappresentare relazioni del tipo N : M.

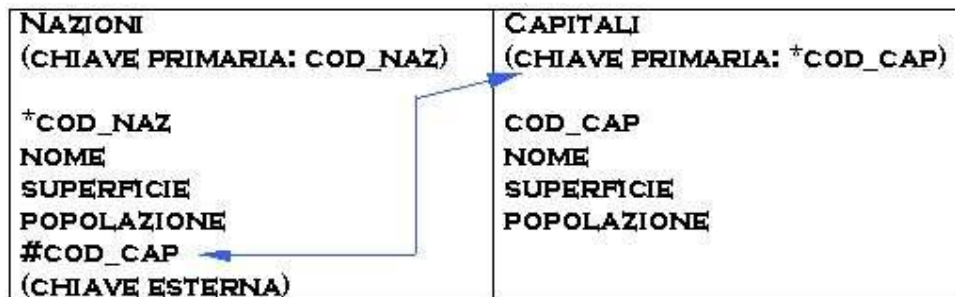
Partendo da uno schema concettuale, si passa alla creazione di uno schema logico relazionale applicando le seguenti regole:

1. *Le entità dello schema concettuale, diventano tabelle.*
2. *Le relazioni tra entità dello schema concettuale, nello schema logico relazionale vengono rappresentate facendo uso di chiavi esterne, ossia un insieme di attributi che corrispondono a quelli che costituiscono la chiave primaria di un'altra tabella, e stabiliscono quindi un riferimento tra le righe delle due tabelle.*

In particolare per rappresentare una relazione tra le due tabelle T1 e T2 si distinguono 3 casi:

1) **Relazione 1: 1**; agli attributi di T1 viene aggiunta, come chiave esterna, la chiave primaria di T2, e viceversa.

Esempio: supponiamo di avere le due entità: nazioni di attributi cod_naz, nome, superficie, popolazione e capitali di attributi cod_cap, nome, superficie, popolazione. La relazione che intercorre tra le due entità è evidentemente 1:1. La traduzione nello schema relazionale avviene creando due tabelle dove una (nazioni per esempio) contiene la chiave primaria dell'altra. In questo modo si è realizzato un riferimento tra la nazione e la sua capitale.



2) **Relazione 1: N**; agli attributi di T2 viene aggiunta, come chiave esterna, la chiave primaria di T1, ma non viceversa.

Esempio: nella relazione 1:n si opera in maniera analoga, ma non è possibile invertire le chiavi esterne. Dalle entità studenti e libri nel Data Base biblioteca si ottengono le seguenti tabelle (la chiave primaria è identificata con * e quella esterna con #).



3) **Relazione N:M**; in questo caso viene definita una nuova tabella, chiamata T3, che contiene, come chiavi esterne, la chiave primaria sia di T1, che di T2, ed eventuali attributi contenuti nella relativa relazione che li lega.

Esempio: la relazione n:m tra autori e libri si realizza invece introducendo un'altra tabella (il nome di quest'ultima è arbitrario) che contiene le chiavi primarie di entrambe le due tabelle precedenti.



PROGETTAZIONE FISICA

In questa fase il problema è individuare la forma migliore in cui salvare i dati che costituiscono gli archivi. Per fare ciò è necessario rispondere a domande come: dove verranno salvati i dati? Che struttura avranno i dati nei file? Che file system si utilizzerà? Quest'ultima fase di solito è realizzata dal DBMS, e dipende, oltre che dal DBMS che è stato scelto, anche dal file system usato, quindi non riguarda il programmatore.

Un semplice esempio di base di dati è l'elenco telefonico. Una pagina di un elenco telefonico cittadino può presentare più o meno i seguenti dati:

Cognome	Nome	Indirizzo	Numero di Tel.
Brambilla	Michele	via delle Puglie 9	0332 7827161
De Magistris	Alessandro	p.zza Matteotti 5	2123 2112231
De Paperoni	Paperon	viale delle Libertà 19	2311 2311231
Rossi	Luca	via Merano 1	3445 7843924

Si noti che i dati dell'esempio sono tutti correlati tra di loro e sono organizzati in ordine alfabetico per facilitarne la ricerca. I titoli delle colonne (Cognome, Nome, Indirizzo e Numero di Telefono) vengono chiamati **metadati**, mentre tutti gli altri sono i dati veri e propri. Ogni "riga di dati" prende il nome di tupla o record o riga, mentre tutto l'insieme forma una **tabella**.

Ora, proviamo ad ampliare l'esempio supponendo di voler costruire una base di dati per la gestione di una rubrica personale. Oltre al nome, al cognome e all'indirizzo di ogni persona vogliamo sapere il posto in cui lavora con relativo indirizzo e recapito telefonico. La tabella diventerebbe:

Cognome	Nome	Indirizzo	Numero di Tel.	Lavoro	Indirizzo Lavoro	Telefono Lavoro
Brambilla	Michele	via delle Puglie 9	0332 7827161	Banca Europe	via del Lavoro 7	099 4567899
De Magistris	Alessandro	p.zza Matteotti 5	2123 2112231	KL Eletttronix	via Nova 134	123 45221123
De Paperoni	Paperon	viale delle Libertà 19	2311 2311231	-	-	-
Rossi	Luca	via Merano 1	3445 7843924	Banca Europe	via del Lavoro 7	099 4567899

Paperon De Paperoni vive di rendita e non ha bisogno di lavorare.

Anche se a prima vista corretta, una memorizzazione del genere non è conveniente: ogni posto di lavoro ha sempre gli stessi recapiti e Brambilla Michele e Rossi Luca lavorano presso la stessa banca. Tale soluzione spreca più spazio di quanto sia davvero necessario e in più ci complica la vita nel caso in cui vogliamo modificare i dati. Infatti se un'azienda cambia numero di telefono dovremo cambiare il numero di telefono in tutte le righe in cui è presente.

Un modo più intelligente di memorizzare tale base di dati è il seguente:



Abbiamo creato due tabelle **associate** tra di loro. Brambilla Michele e Rossi Luca saranno associati entrambi alla riga relativa a Banca Europe, De Magistris Alessandro sarà associato alla riga della KL Elettronix, Paperon. De Paperoni invece non sarà associato a nessun posto di lavoro. Si noti che adesso un'eventuale modifica ai dati di un posto di lavoro si rifletterà in modo automatico e trasparente su tutte le persone che vi lavorano.

Da questo semplice esempio è evidente come sia importante progettare nel modo corretto una base di dati. L'organizzazione dei dati ha un'importanza cruciale che si ripercuote sulla semplicità di gestione e sulle prestazioni del sistema. Se la progettazione viene fatta in modo sbagliato o, peggio ancora, non viene per nulla effettuata si rischia di ottenere un database di difficile utilizzo e che farà sicuramente perdere molto tempo nella stesura delle funzioni per la manipolazione dei dati.