

# Sviluppo di Sistemi Informativi - Università di Pavia

prof. Ruggero Ferrari

## INTRODUZIONE

### I presupposti di uno sviluppo industriale dei sistemi informativi

Introduzione

Piattaforme HW e SW standard

I fattori critici di successo

Le strutture fondamentali e ricorrenti: il livello applicativo

### Analisi implementativa di una relazione mono tabellare

...

...

..

## Relazione Master-Detail

...

...

..

..

## l'implementazione delle tabelle di movimentazione (storici)

..

..

..

..

### *referimenti:*

- Oracle Developmen Tools
- WEBDB - HTMLDB
- DBDesigner4 - Umbrello - MR Planner

## INTRODUZIONE

*Dai discorsi riguardanti la sicurezza dei sistemi ricaviamo un grande insegnamento:*

***La maggioranza dei problemi del software e dei sistemi operativi nasce dal scarsa qualità dello sviluppo del codice e delle incoerenze tra l'analisi e lo sviluppo.***

*Conseguentemente a questa considerazione possiamo formulare un obiettivo fondamentale e guida del lavoro che costituirà il nostro approfondimento futuro:*

Progettare, sviluppare e mantenere un sistema basato su standard solidi (di qualità) e ad alto impatto industriale (ridotto intervento umano) può costituire la base per avere prototipi e rilasci di software e sistemi SICURI

**standard sull'HARDWARE:**

piattaforme con un ampio campo di scalabilità e compatibilità di Sist. Operativi

**standard sui SISTEMI OPERATIVI:**

prodotti con un ampio campo di compatibilità sulle piattaforme di connettività e sviluppo

**standard sulle RETI:**

architetture con un ampio campo di compatibilità sui driver di scambio file e applicazioni di di groupware (emulazioni, multichannel, signal exchange, LAN e WAN connection, ...)

**standard sui RDBMS:**

DataBase migrabile e scalabile di piattaforma Hardware e Software di Base. Capacità di scambio dati con standard come XML o CVS verso e da altre fonti. SQL integrato.

**standard sull SOFTWARE:**

Linguaggi facilmente integrabili nel contesto di gestione dati (SQL) e presentation(HTML).

**standard sull' INTERFACCIA UTENTE:**

Applicabile ad un Browser (HTML o Java) com e standalone (Client-Server)

**Le strutture fondamentali e ricorrenti:  
Il livello applicativo**

**GESTIONE STATICA**

**A. Gestione mono tabellare (mono relazione)**

Esempio: Codice Materiale con i suoi attributi come Unità di misura, descrizione,...

**B. Gestione Master-Detail (testata e righe)**

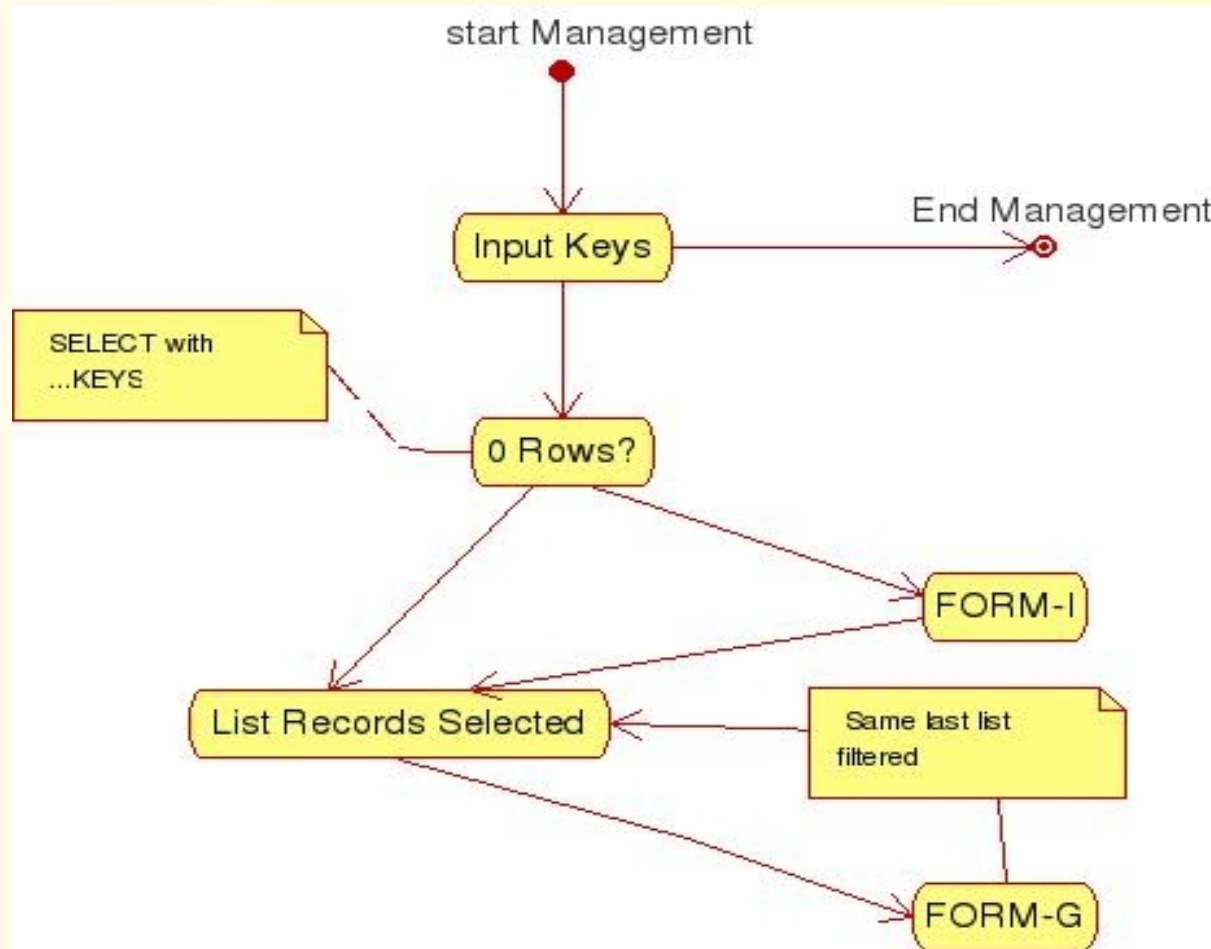
Esempio: Info di Testata ordine (data ordine, cliente, ...) e righe di dettaglio collegate (codice materiale, quantità ordinata, prezzo concordato, sconto, ...)

**GESTIONE DINAMICA**

**C. Gestione storico o movimentazione**

Esempio: Carico e scarico di materiali a magazzino caratterizzato da grandi volumi e da un frequente utilizzo. E' necessario verificare l'ottimizzazione dei percorsi dell'utente ed è quindi il più difficilmente "industrializzabile".

# Flusso di gestione di una relazione Monolitica



## Gli elementi:

Ricerca con chiavi significative e report attivo dei risultati

FORM di Inserimento record nuovo (solo KEY?)





FORM di Gestione di un record esistente (compresa cancellazione)

Workflow Standard

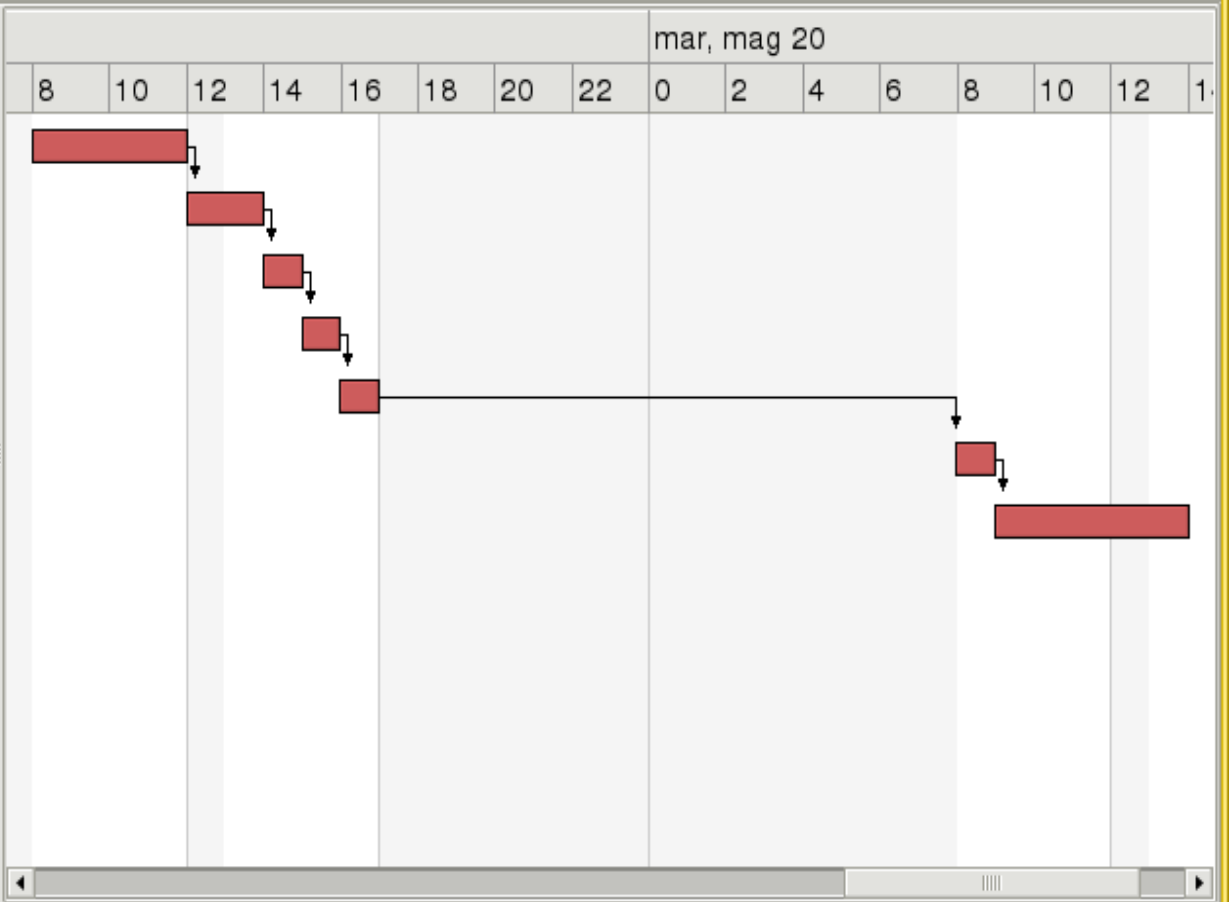
GANTT-Std-MonoTAB.planner - Planner

File Modifica Vista Azioni Progetto Aiuto



-  Diagramma di Gantt
-  Attività
-  Risorse
-  Utilizzo risorse

WBS	Nome	Lav
1	Analisi	4h
2	Progetto Tabella	1h
3	Produco FORM-I	1h
4	Produco FORM-G	1h
5	Produco Report c	1h
6	Replico WF	1h
7	Test di Gestione	4h



## **POSSIBILI COMPLESSITA' POSSONO CREARE “FUORI STANDARD”**

- **Recupero dati**
- **Validazione campi complessa**
  - **Granularità degli input discreti (rosa di valori definita)**
  - **Supporto all'input strutturato (date, Pop-up, combo, ...)**
  - **Inserimento del nuovo elemento nelle scelte discrete**  
(link alla FORMI della tabella relativa)

ANALISI: fino ai dettagli implementativi

- **Obiettivi , Vincoli e Specifiche**  
(Cosa, a quali Condizioni, Come)
- **Case Diagram**  
(Attori e Funzionalità)
- **State Diagram**  
(transazioni di Processo)
- **ER Diagram**  
(definizione entità e relazioni)



# ● Obiettivi , Vincoli e Specifiche (Cosa, a quali Condizioni, Come)

E' sufficiente un blocco anche cartaceo per raccogliere durante un colloquio con l'utente i requisiti dello sviluppo. In questo momento si gioca il massimo dell'impatto qualitativo di progetto: un'incomprensione di termini, una mancanza di chiarimenti sui bisogni reali, un sottinteso non esplicitato può provocare il rifacimento completo del lavoro di sviluppo.

Come già citato nel terzo modulo del corso di sistemi informativi 1 (terzo anno) l'esperienza nel condurre il dialogo con l'utente applicando tecniche "euristiche" per indurre ad esplicitare i requisiti in forma completa risulta determinante.

Esempio:

COSA:

Vorrei non avere ritardi di produzione a causa della mancanza di materiale a magazzino

A QUALI CONDIZIONI:

Semplicità di gestione (a prova di magazziniere)

Costo contenuto

Integrazione con la gestione ordini dell'ufficio acquisti

COME:

Registrazione arrivi a magazzino e registrazione scarichi per utilizzo in produzione

Analisi delle scorte necessarie con interfaccia "intelligente" (algoritmo di calcolo automatico)

Analisi dei riordini necessari con una proposta d'ordine.

# ● Case Diagram (Attori e Funzionalità)

## USE CASE DIAGRAM

Fa vedere cosa l'applicazione può fare; è uno dei diagrammi più di alto livello.  
Il primo passo di "qualsiasi" processo di sviluppo è la definizione dei requisiti. Occorre avere le idee chiare su cosa l'applicazione deve fare.

Gli USE CASE:

- definiscono il comportamento del sistema (cosa deve fare e non come)
  - o Le funzionalità (processi) principali
  - o Come il sistema agisce e reagisce (ambiente ↔ utenti)
- descrivono
  - o Il sistema
  - o L'ambiente
  - o Le relazioni fra sistema e ambiente
  - o ogni diagramma può avere diversi livelli di dettaglio

*Definisco cosa il sistema offre (servizi per l'utente finale...)*

*Non definisco come il sistema è realizzato (implementazione, dettagli interni del sistema, architettura, topologia dell'applicazione...)*

## ELEMENTI GRAFICI

ACTOR : è qualcuno (utente) o qualcosa (sistemi esterni - hardware) che:

- Controlla le funzionalità
- Fornisce input o riceve output dal sistema

Non fa parte del sistema da sviluppare

Rappresenta una classe di utenti e non un singolo utente



Identificati gli actor, vediamo quali sono le funzionalità, cosa l'applicazione deve fare.

Gli use case possono essere ricavati dalle interviste con gli utenti.

Identificano:

- Gli obiettivi: ciò che il sistema dovrebbe fare secondo gli utenti
- Le interazioni: cosa vorrebbero (potrebbero) fare i diversi utenti e come dovrebbe interagire il sistema con gli altri già presenti

Gli use case di alto livello sono volutamente generici

- I dettagli vanno aggiunti raffinando le funzionalità del sistema
- Bisogna esplorare le diverse possibilità per non introdurre prematuramente scelte progettuali

### **FUNZIONALITA' FONDAMENTALI**

Il processo di definizione degli use case è iterativo

- Si inizia identificando il comportamento più semplici
- Si descrivono i comportamenti alternativi e più complessi

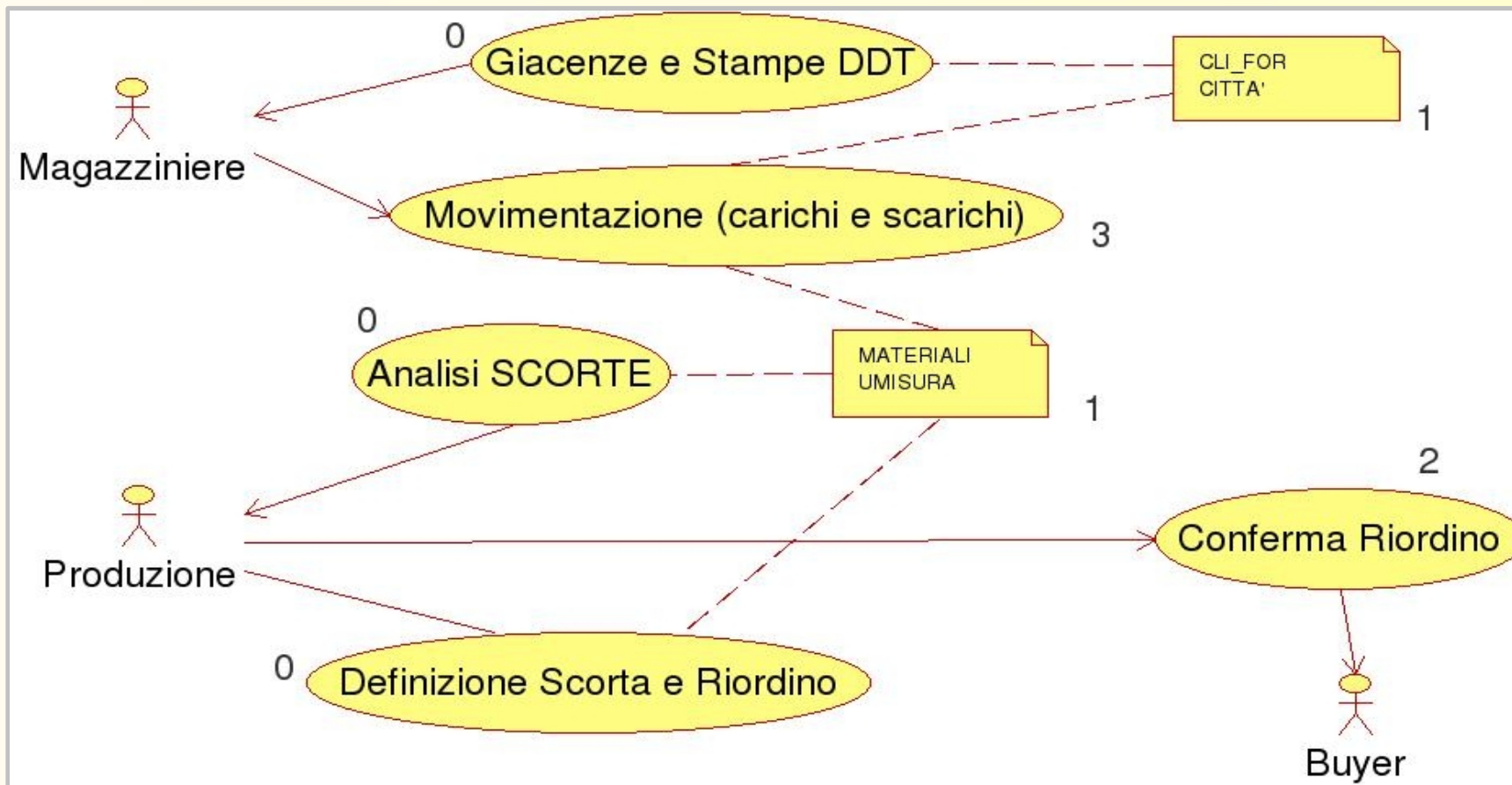
Quando smettere?

- Un buon livello di dettaglio facilita tutte le attività successive
- Troppi dettagli: complicherebbero inutilmente la descrizione, introdurrebbero prematuramente scelte progettuali, e precluderebbero la visione d'insieme

### **COSA NON SI PUÒ FARE**

- Non ha senso collegare due actor con un association
- Se due utenti parlano fra loro, questo non fa parte dell'applicazione
- Non ha senso collegare due use case con un association (La comunicazione all'interno dell'applicazione è un "dettaglio implementativo" e non va descritto nei requirement)

## Esempio di Diagramma dei casi d'uso



Sono evidenziati gli elementi esterni alle funzioni come gli attori (Magazziniere, Buyer, ...) e le entità di Data Base (Materali, Clienti\_Fornitori, Città,...).

I numeri a fianco degli elementi grafici indicano la natura degli stessi:

0 --> Report

1 --> Gestioni Mono Relazionali

2 --> Gestioni Master Detail

3 --> Gestioni Movimentazione

# ● State Diagram (transazioni di Processo)

Descrive il comportamento di un attività (Processo).  
Sono in grado di dire come un oggetto reagisce quando riceve un segnale oppure in base ad un evento mediante azioni e transizioni.

## ELEMENTI GRAFICI

Inizio:

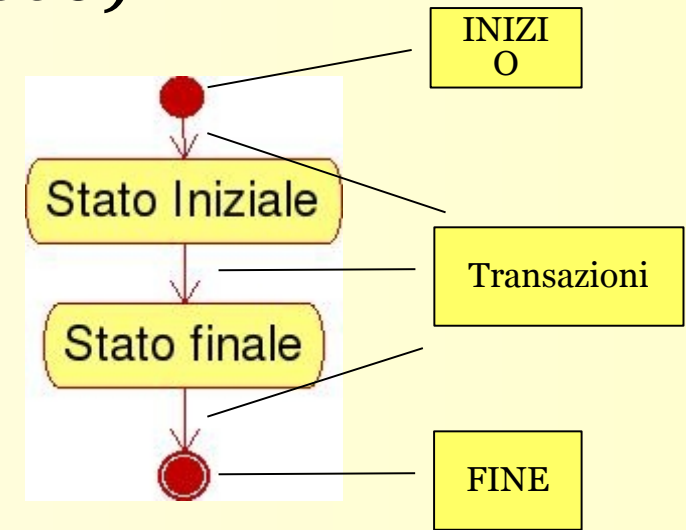
Fine:

Stato intermedio:

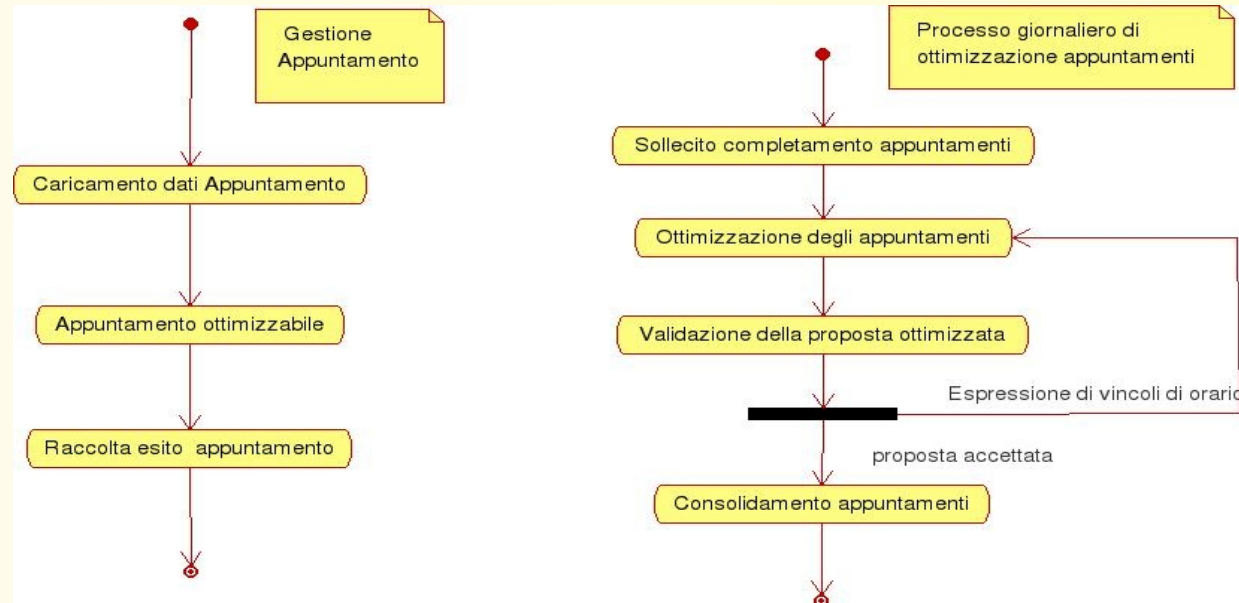
Transizione: in certe condizioni / esegue azioni

Se le condizioni sono vere,

==> la transizione scatta e vengono eseguite le azioni



## ESEMPIO:



# ● ER Diagram (definizione entità e relazioni)

Gli strumenti industriali per la progettazione e la produzione del codice prevedono un supporto per la fase di analisi del modello LOGICO, guidano la transazione dal Modello CONCETTUALE a quello LOGICO.

Prodotto un efficace modello LOGICO effettuano la produzione automatica o semiautomatica del modello FISICO.

*Per avere un riferimento OpenSource come esempio di questa categoria di strumenti, si può far riferimento a : [www.fabforce.net](http://www.fabforce.net) con DBDesigner4*

