PROGETTAZIONE CONCETTUALE

Analisi

Si parte da una **realtà d’interesse** (**traccia** del problema) che va letta e capita in modo da riconoscere le entità.

Se questa dice di gestire i parcheggi nella città di Varese, la realtà d’interesse è Varese, quindi diverso da magari Milano. IMPORTANTE quindi CIRCOSCRIVERE IL PROBLEMA. Seguono **3 fasi** di progettazione:

**Progettazione concettuale**

Consiste nell’**organizzare** tutto **ciò** **che** **si** **ha** **a disposizione** (dalle specifiche di un problema) **per creare** **un** modello astratto o **schema concettuale** **del db** che **rappresenti la realtà di interesse** indipendentemente dal DBMS (**concettuale** perché **evita** il + possibile di **descrivere dettagli di realizzazione**, **ma** **solo** la **struttura** dei **dati**).

Input --> Specifiche del problema (traccia).

Output --> ER Diagram.

**Progettazione logica**

Trasforma lo **schema concettuale** in uno **schema logico**, 1 rappresentazione efficiente rispetto a strutture del DBMS.

Input --> ER Diagram.

Output --> Schema logico riassumibile con relazioni rappresentate da tabelle logiche.

**Progettazione fisica**

**Implementa** lo **schema logico** definendo tutti gli aspetti fisici di memorizzazione e rappresentazione **in memoria**.

Input --> Schema logico (tabelle logiche) trovate con la progettazione logica.

Output --> Implementazione in memoria delle tabelle.

ENTITA

Entità

Sono **oggetti della realtà** **raggruppati** **in** **una** **classe** **dato che hanno caratteristiche comuni** (elementi della realtà di cui è **necessario** il **salvataggio** di **dati**. Rappresentate con un **rettangolo**. Sempre **nomi singolari** + **iniziale maiuscola**.

Istanza

Ogni singolo **esemplare che appartiene a un** certo tipo di **entità**.

Attributi

**Proprietà con cui è descritta un’entità**. Ogni attributo ha un nome. Dati calcolati NON sono attributi (tipo l’età). Rappresentati con **linea** (che parte da entità) e **pallino** (pieno o vuoto) con **nome** attributo. Possono essere:

* **Obbligatori** --> se **NOT NULL**, tipo un nome;
* **Facoltativi** --> se **accettano** valori **NULL**; rappresentati con (0,1 singoli) o (0,N multivalore) sull’attributo.
* **Semplici** --> unici e indivisibili;
* **Composti** --> costituiti con **aggregazione** di **altri attributi**;
* **Multivalore** --> **accettano** **valori** **multipli**, tipo liste;
* **Derivati** --> **calcolati** e ottenuti da elaborazioni;
* **Chiave** --> **identificano univocamente l’istanza**, un attributo chiave può essere:
  + **Candidata** --> **Insieme di 1 o + attributi** (singola o no) che identificano univocamente un’istanza

Minimalità

Le chiavi devono rispettare questo principio, quindi una **chiave candidata deve contenere il < n° di valori possibili** **per identificare univocamente le istanze**. Se non so scegliere la PK uso un attributo CODICE.

Un attributo è composto da:

**Nome**, **formato** (tipo di valori che può assumere), **dimensione** (qta max di char o max int length) e **valore**.

Dominio

L’**insieme di possibili valori** (omogenei) **assunti da un attributo**.

Gerarchie

Accadono negli ER dei casi di **ereditarietà tra entità padri e figlie**. Una **gerarchia** ha una struttura ad **albero**, dove la **root** è **l’entità con gli attributi comuni** tra i figli, e i **figli hanno attributi “privati” validi per loro**.

Meglio non creare gerarchie siccome poi vanno risolte.

Vincoli di integrità

**Regole da rispettare durante** le operazioni di **INSERT**, **UPDATE** o **DELETE** dei dati affinché il **db** **rimanga** **consistente**.

Impongono restrizioni sui valori assumibili dagli attributi (tipo età negativa o data assunzione prima di data di nascita).

Tipo Auto è posseduta da Proprietario (per targa), se cancello Proprietario rimane Auto senza Proprietario. Divisi in:

Espliciti

Tipo: V[*N*]: ([Entità.Attributo] [condizione/espressione]). ES: “V1: (…DataNascita < …DataAssunzione)”.

Impliciti

Risultano dall’ER, non si fa nulla:

* **Vincolo di PK** --> necessaria una PK.
* **Vincoli di FK** --> ES: tra entità A e B dove c’è FK, non si può mettere un elemento in A senza anche metterlo in B.

ASSOCIAZIONI

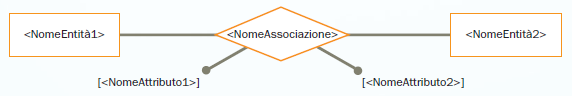
Associazione

(O relationship) è una **relazione logica tra 2 o + entità rilevanti della realtà**.

Istanza

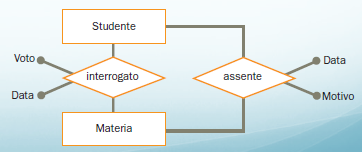
Di una associazione è un **fatto che descrive un’azione o una situazione e crea relazioni tra le istanze di entità**.

Graficamente:



(Di solito: sostantivi = entità, verbi = associazioni, e tra 2 entità ci possono essere + associazioni)

Le **associazioni possono avere** **attributi**, consideriamo il caso di Studente e Materia.



Grado

È il n° di entità che sono nell’associazione. 2 entità = associazione di grado 2 o binaria. Le associazioni di grado > di 2 si dicono multiple, e vanno riportate a grado 2. Esempio:

|  |  |
| --- | --- |
|  |  |

Tipi di associazioni

Ricorsiva

È una associazione di grado 1, quindi da sola su sé stessa. È detta ricorsiva.

Però non si sa in che relazione siano le entità tra loro stesse, quindi va inserito il ruolo delle varie entità tra loro stesse nelle associazioni.

Diretta

È l’associazione tra entità 1 e entità 2 (*verbo attivo*)

Inversa

È l’associazione tra entità 2 e entità 1 (*verbo passivo*)

Totale

Quando ad ogni elemento di entità 1 deve corrispondere almeno 1 elemento di entità 2.

Parziale

Quando può esistere un elemento di entità 1 a cui non corrisponde alcun elemento di entità 2 o viceversa.

Tipi di associazioni

Molteplicità

La molteplicità di un’associazione tra 2 entità indica il **n° massimo di istanze dell’entità 1 che si possono trovare in relazione con un’istanza dell’entità 2**. In base a questa le associazioni possono essere tipo:

1:1 (Uno a uno)

Quando un’istanza di entità 1 corrisponde a una sola istanza di entità 2 e viceversa. Le 1 a 1 sono **rare**. Graficamente:

(img)

1:N (Uno a molti)

Quando a un’istanza di entità 1 possono corrispondere + istanze di entità 2, ma ogni istanza di entità 2 ha 1 singola entità 1. Può essere anche al contrario, ma è solo in un senso. Graficamente:

(img)

N:N (Molti a molti)

Quando a ogni istanza dell’entità 1 possono corrispondere 1 o + istanze dell’entità 2 e viceversa. Graficamente:

(img)

ASSOCIAZIONI VANNO SPIEGATE, DICENDO RAGIONAMENTO LOGICO E PERCHE SCELTO QUEL TIPO DI MOLTEPLICITA

Cardinalità

La cardinalità di un’associazione definisce il **n° di istanze con cui l’entità può partecipare in un’associazione**.

Con questa è possibile trovare il n° minimo e il n° massimo di istanze che partecipano all’associazione.

È identificata da 2 numeri tra parentesi tonde di fianco all’entità che definiscono minimo e massimo.

Esempio 1

Medico --> visita --> Paziente

Min pazienti visitabili da Medico --> 1

Max pazienti visitabili da Medico --> ? --> N

Min medici che possono visitare un Paziente --> 1

Max medici che possono visitare un Paziente --> ? --> N

Esempio 2

Squadra --> gioca --> Giocatore

Qui non posso usare la molteplicità, siccome il n° minimo di giocatori in una squadra è 5.

Differenze

Una cardinalità minima = 0 indica che l’entità partecipa in modo opzionale all’associazione.

Molteplicità --> Persona - - 1 - - possiede ---N--- Auto

Cardinalità --> Persona – (0,N) – possiede – (1,1) - Auto

Attenzione

La molteplicità deve esplicitare associazioni parziali e il numero va messo in mezzo all’associazione.

La cardinalità deve avere i numeri vicino all’entità a cui si riferisce.

La cardinalità è obbligatoria quando nella realtà d’interesse ci sono dei vincoli (min o max) sul numero di istanze delle entità dell’associazione.

Ristrutturare un ER

Quando si fa un ER, alla fine è sempre meglio fare un check di ste cose:

Eliminare attributi composti

Attributi composti non sono rappresentabili nei DBMS, quindi vanno portati ad attributi semplici (1 per ogni colonna).

Eliminare attributi multivalore

Per rimuoverli, si crea una nuova entità contenente i valori dell’attributo, e collegandola con una relazione 1 – N.

Tipo se una persona ha + numeri di telefono, si crea una associazione persona --> recapito --> telefono (con numero).

Eliminare gerarchie

RDBMS non possono rappresentare gerarchie, ma solo entità e relazioni; quindi queste vanno risolte. 3 modi:

Accorpamento di figli nel padre

Esempio: Persona (nome, cognome) <-- Docente (materie, classe)

Persona incorpora gli attributi del figlio, con: (nome, cognome), materie [0,N], classe [0,1].

Svantaggio: se ho tante sottoentità si ha un padre con troppi attributi NULL.

Accorpamento del padre nei figli

Esempio: Persona (nome, cognome) <-- Docente (materie, classe)

Figli incorporano gli attributi del padre, con: nome, cognome, materie, classe.

Modo più semplice per risolvere.

Sostituzione di gerarchia con entità e associazioni

Esempio: Persona (nome, cognome) <-- Docente (materie, classe)

Si mette una relazione 0 – 1 e 1 – 1 --> Persona – (0,1) – è – (1,1) - Docente

Considerare attributi che diventano entità

Esempio: Insegnante (nome, scuola). Scuola non può essere un attributo semplice, siccome sul territorio sono quelle.

O stessa cosa con paesi. Non cambiano, o raramente lo fanno. Quindi per non ripetere sempre il paese o la scuola in ogni record, metto tutti i paesi o scuole in tabella e passo solo la FK, così da evitare ridondanze e inconsistenze.

Controllare associazioni 1 – 1

PROGETTARE UN ER

Passi

1) Analizzare specifiche formali considerando tutti gli aspetti.

2) Elencare eventuali ipotesi aggiuntive/semplificative (tipo si suppone che un brano possa essere scritto da + autori) in chiaro, per spiegare una scelta o semplificarci la vita (non subito all’inizio, lasciare spazio).

3) Analisi: descrivere (circa mezza pagina) che spiega a chi lo legge il progetto da fare (tipo di info da memorizzare, soluzioni da adottare…), cosicché chi legga il tema capisca che si stanno facendo certe cose per un motivo (non manuale, tipo un tema che spiega progetto, con cause conseguenze e perché).

4) Elencare le entità individuate nella realtà di interesse spiegando perché e a che serve (si lega bene con punto 3, perche magari si scrive: “usando le entità: libro, per …; autore, per…”. Sono anche insieme, ma non necessario).

5) Fare il dizionario dei dati. Roba fondamentale (Nome, Tipo, Dimensione, Obbligatorio si o no, PK si o no). Si può fare anche in forma testuale tipo la 3, ma non consigliato.

6) Elencare eventuali vincoli, in formato: Vincolo n° [N] (… > …) AND/OR (… < …)

7) ER

8) Spiegare le associazioni (perche cardinalità o perche N a N)

ESERCIZIO 1

Agenzia immobiliare, definire entità, attributi con dizionario dei dati (tabella con colonne: nome, tipo, dimensione, PK?, O/F (obbligatorio o facoltativo), descrizione), vincoli e ER.

Immobili in vendita: identificati da un codice, interessa il tipo (appartamento, villa, …) superficie, n° stanze, annessi (garage, cantina, giardino…) prezzo e proprietario.

Proprietario ha: codice fiscale, cognome, nome, telefono, immobili in vendita di cui è proprietario.

ENTITA’

Immobile:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
| Codice | Varchar | 10 | X | X | Codice immobile |
| Tipo | Varchar | 25 |  | X | Tipo immobile |
| Superficie | Decimal | 11, 2 |  | X | Superficie |
| N\_stanze | Int | 11 |  | X | Numero stanze |
| Prezzo | Decimal | 11, 2 |  | X | Prezzo |
| Proprietario | Int | 11 |  | X | Proprietario |

Vincoli:

V1: Immobile.Tipo IN (“Appartamento”, “Baita”, “Villa”)

V2: Immobile.Superficie > 0

V3: Immobile.Stanze > 0

V4: Immobile.Prezzo >= 0

Proprietario:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
| CodFis | Varchar | 16 | X | X | Codice Fiscale |
| Nome | Varchar | 40 |  | X | Nome prop |
| Cognome | Varchar | 40 |  | X | Cognome prop |
| Telefono | Varchar | 11 |  | X | Telefono prop |

ESERCIZIO 2

Gestire archivio film on entità, tabella data dictionary, vincoli, ER

ENTITA’

Supporto ???

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
| Codice | Varchar | 10 | X | X | Codice supporto |
| TitoloFilm | Varchar | 50 |  | X | Id film |
| Posizione | Int | 11 |  | X | Posizione |

Vincoli

V1: Supporto.Posizione > 0

ER

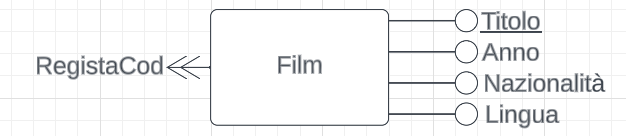
Film

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
| Titolo | Varchar | 50 | X | X | Codice supporto |
| Anno | Int | 11 |  | X | Id film |
| Nazionalità | Varchar | 30 |  | X | Nazionalità |
| Lingua | Varchar | 30 |  | X | Lingua |
| RegistaCod | Varchar | 10 |  | X | Codice regista |

Vincoli

V1: Film.Anno < YEAR(CURRENT\_DATE())

ER



Attore

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
| Codice | Varchar | 10 | X | X | Codice |
| Nome | Varchar | 40 |  | X | Nome |
| Cognome | Varchar | 40 |  | X | Cognome |
| DataNascita | Date |  |  | X | DataNascita |
| LuogoNascita | Varchar | 40 |  | X | LuogoNascita |
| Foto | BLOB |  |  | X | Foto |

Vincoli

V1: Attore.DataNascita < CURRENT\_DATE()

ER



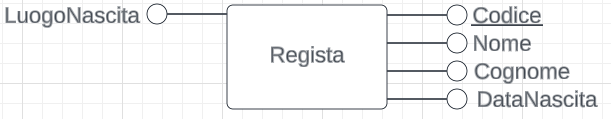
Regista

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
| Codice | Varchar | 10 | X | x | Codice |
| Nome | Varchar | 40 |  | X | Nome |
| Cognome | Varchar | 40 |  | X | Cognome |
| DataNascita | Date |  |  | X | DataNascita |
| LuogoNascita | Varchar | 40 |  | X | LuogoNascita |

Vincoli

V1: Regista.DataNascita < CURRENT\_DATE()

ER



ESERCIZIO 3

Teatro verdi deve gestire info per stagione lirica. Stagione lirica ha + spettacoli. Spettacolo è rappresentazione di un’opera. Spettacolo: regista, direttore, interpreti, date. Opera: titolo, autore libretto, autore musica, anno, luogo 1° rappresentazione. Interpreti sono cantanti, di loro: nome, personaggio interpretato da ogni cantante in ogni spett.

ENTITA’

Spettacolo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
| Codice |  |  |  |  |  |
| Date |  |  |  |  |  |
|  |  |  |  |  |  |

Vincoli

V

ER

Opera

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Vincoli

V

ER

Regista

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Vincoli

V

ER

Direttore

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Vincoli

V

ER

Interprete

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Vincoli

V

ER

Data di rappresentazione ???

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOME | TIPO | DIMENSIONE | PK | NOT NULL (obb.) | DESC |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Vincoli

V

ER