CAPITOLO 1: PROGETTAZIONE CONCETTUALE

1.1 I modelli concettuali dei dati

- **Modello =** insieme di regole e strutture che permettono la rappresentazione della realtà di interesse
- **Schema =** la rappresentazione di una specifica realtà secondo un determinato modello

OSS:

1) il modello fornisce le regole per la costruzione della rappresentazione

2) la rappresentazione è data da un insieme di simboli posti in corrispondenza con la realtà di interesse

3) la realtà di interesse è una porzione di mondo reale così com'è percepita da chi costruisce la rappresentazione

- **Astrazione:** procedimento mentale che si adotta quando si evidenziano alcune proprietà e caratteristiche di un insieme di oggetti

PRIMITIVE DI ASTRAZIONE:

A) CLASSIFICAZIONE: definizione di una classe a partire da un insieme di oggetti caratterizzati da proprietà comuni

Relazione: MEMBER_OF --> i membri della classe SETTIMANA sono Lunedì, … , Domenica

B) GENERALIZZAZIONE: definisce una relazione di sovra-insieme tra una classe padre e altre classi figlie

Relazione: IS_A --> la classe PERSONA è una generalizzazione delle classi UOMO e DONNA

La generalizzazione stabilisce un mapping tra gli elementi di una classe egli elementi delle classi generalizzate. Per tale mapping si definiscono le proprietà di copertura:

- **Copertura totale o parziale:**

Totale (t): ogni elemento della classe generalizzazione è in relazione con almeno un elemento delle classi generalizzate;

Parziale (p): esistono alcuni elementi della classe generalizzazione che non sono in relazione con alcun elemento delle classi generalizzate;

- **Copertura esclusiva o sovrapposta:**

Esclusiva (e): ogni elemento della classe generalizzazione è in relazione con al massimo un elemento delle classi generalizzate;

Sovrapposta (o): esistono alcuni elementi della classe generalizzazione che sono in relazione con elementi di due o più classi generalizzate

C) AGGREGAZIONE: permette di giungere alla definizione di un concetto a partire da altri concetti che ne rappresentano le parti

Relazione: PART_OF --> il concetto di BICICLETTA è la classe i cui componenti sono RUOTA, PEDALE e MANUBRIO

Stabilisce una corrispondenza tra gli elementi di una classe componente e gli elementi della classe composta

- Aggregazione binaria: è una corrispondenza stabilita tra due classi
- Aggregazione n-aria: è stabilita tra tre o più classi C1, C2, … , Cn

Cardinalità minima di Ci in A: è il minimo numero di corrispondenze nell'aggregazione A alle quali ogni membro di Ci *deve* partecipare

Cardinalità massima di Ci in A: è il massimo numero di corrispondenze nell'aggregazione A alle quali ogni membro di Ci *può* partecipare

CARDINALITA' della partecipazione alla corrispondenza

- Sia A un'aggregazione di C1 e C2:

Cardinalità minima di C1 in A: è il minimo numero di corrispondenze nell'aggregazione A alle quali ogni membro di C1 *deve* partecipare

Cardinalità massima di C1 in A: è il massimo numero di corrispondenze nell'aggregazione A alle quali ogni membro di C1 *può* partecipare

Partecipazione opzionale: alcuni elementi di C1 possono non essere aggregati tramite A elementi di C2

Partecipazione obbligatoria: ad ogni elemento di C1 deve essere aggregato, tramite A, almeno un elemento di C2

Cardinalità della corrispondenza:

- Uno a uno:

Max-card(C1,A) = 1

Max-card(C2,A) = 1

- Uno a molti:

Max-card(C1,A) = n

Max-card(C2,A) = 1

- Molti a uno:

Max-card(C1,A) = 1

Max-card(C2,A) = n

- Molti a molti:

Max-card(C1,A) = n

Max-card(C2,A) = n

1.2. Il modello Entity Relationship (E/R)

Elementi base:

- **Entità:** rappresenta un insieme di oggetti della realtà di cui si individuano proprietà comuni
- **Associazione:** rappresenta un *legame logico tra due o più entità* --> *sono caratterizzate in termini di cardinalità*
- **Attributo:** *proprietà elementari* di entità o associazioni

Dominio dell'attributo: insieme di valori legali per l'attributo.

Attributo Semplice = se è definito su un unico dominio

Attributo Composto = è costituito da un gruppo di attributi che hanno affinità nel significato o nell'uso

ATTRIBUTO VS ASSOCIAZIONE (PG. 20-21)

OSS: Sia E un'entità o un'associazione e A un attributo di E:

Cardinalità minima: è il minimo numero di valori dell'attributo A associati ad ogni istanza dell'entità o associazione E

Cardinalità massima: è il massimo numero di valori dell'attributo A associati ad ogni istanza dell'entità o associazione E

Cardinalità opzionale: min-card(A,E) = 0 --> può non essere specificato il valore dell'attributo

Cardinalità obbligatoria: min-card(A,E) = 1 --> almeno un valore dell'attributo deve essere specificato

Valore-singolo: max-card(A,E) = 1

Valore-multiplo: max-card(A,E) > 1

- **Anelli:** è un'associazione binaria tra un'entità e sé stessa. Il ruolo dell'entità è indicato tramite una *label.*
- **Subset:** è una gerarchia di generalizzazione con la sola entità generalizzata

Gerarchie di generalizzazione: un'entità E è una generalizzazione di un gruppo di entità E1, …. , En se ogni oggetto delle classi E1, ... , En è anche oggetto della classe E.

Proprietà di copertura della generalizzazione:

- Totale ed esclusiva
- Totale e sovrapposta
- Parziale ed esclusiva
- Parziale e sovrapposta

Ereditarietà delle proprietà: nell'astrazione di generalizzazione tra tutte le proprietà dell'entità generalizzazione sono ereditate delle entità generalizzate. Nel modello E/R ogni attributo, associazione o generalizzazione definita per l'entità generalizzazione è ereditata automaticamente da tutte le entità generalizzate E1, … , En.

Identificatori

Un identificatore di un'entità E è una collezione di attributi e/o entità in associazione con E che individua in modo univoco tutte le istanze di E.

Definizione formale di identificatore --> Sia E un'entità e siano:

A1, … , An: attributi a valore singolo ed obbligatorio per E

E1, .. , Em: entità diverse da E e connesse ad E tramite associazioni binarie R1, … , Rm obbligatorie e one.to.one o many-to-one.

OGNI ENTITA' DEVE AVERE ALMENO UN IDENTIFICATORE!

Entità forte: ha solo un identificatore interno

Entità debole: ha solo identificatori esterni

1.3. I vincoli di integrità

Un vincolo di integrità *è una condizione che deve essere soddisfatta dalle istanze di una base di dati*. I vincoli di integrità esprimibili variano in base al modello o linguaggio utilizzato per descrivere i dati.

Nel modello E/R i vincoli di integrità esprimibili possono essere classificati nelle seguenti categorie:

- **Vincoli di cardinalità:** per le associazioni e gli attributi
- **Vincoli di copertura:** per le generalizzazioni
- **Vincoli di identificazione:** per le entità

OSS: il concetto di vincolo di integrità mette in evidenza il **ruolo duale di un identificatore:**

A) *Serve per identificare in maniera univoca le istanze di un'entità*

B) *Impone condizioni sulle istanze di un'entità*

1.3.1. La reificazione:

Si intende la rappresentazione di un'associazione in una forma equivalente tramite un'entità:

Sia A un'associazione tra E1, .. , En. La reificazione di A è un'entità *(A-REIFICATA),* collegata a ciascuna delle entità Ei che partecipano all'associazione A, tramite un'associazione binaria molti-a-molti A_Ei con: *card(A,A_Ei) = (1, 1) e card(Ei,A_Ei) = card (Ei,A).*

- **Come definire l'identificatore dell'associazione reificata?**

  Sia A un'associazione tra E1, .. , En. Ogni Ei tale che max-card(Ei,A) = 1 è un identificatore di A-Reificata; altrimenti quando la cardinalità è maggiore di 1, A ha in unico identificatore costituito dalle entità E1, .., En.

1.7 Documentazione di schemi E/R

Uno schema E/R può essere uno strumento incompleto per rappresentare tutti gli aspetti e i vincoli di un dominio applicativo, per varie ragioni:

1) i nomi delle entità e attributi possono non essere sufficiente per comprendere il significato di quello che rappresentano;

2) vari tipi di vincoli di integrità non possono essere espressi direttamente dai costrutti del modello

- **Documentazione di schemi E/R:** uno schema E/R è corredato con una documentazione di supporto allo *scopo di facilitare l'interpretazione dello schema stesso e di descrivere vincoli di integrità non esprimibili in E/R*
- **Regole aziendali (business rules):** *la descrizione di una proprietà che non è possibile rappresentare direttamente nel modello concettuale* può essere espressa mediante delle regole aziendali. *Una regola aziendale può essere:*

A) una descrizione di un concetto (entità, associazione, attributo) dello schema E/R ---> l'insieme delle regole di questo tipo viene identificato come **dizionario dei dati;**

B) un vincolo di integrità, *sia esso la documentazione di un vincolo* espresso nello schema E/R *o la descrizione di un vincolo non esprimibile in E/R;*

C) una derivazione ovvero *un concetto che può essere ottenuto attraverso un'interferenza o un calcolo da altri concetti dello schema* **(dato derivato).**

Oss: non esiste un formalismo standard per esprimere le regole aziendali, ma normalmente si usano definizioni in linguaggio naturale opportunatamente strutturate. Una metodologia condivisa di documentazione di schemi E/R definisce il dizionario dei dati attraverso due tabelle che rappresentano rispettivamente le entità e le associazioni. Le regole aziendali, invece, vengono espresse in un'altra tabella nella quale si indicano le regole specificando di volta in volta la tipologia di appartenenza.

-->> **Quando lo schema concettuale viene tradotto in schema logico e quindi in SQL, le regole aziendali devono essere opportunatamente implementate. Esistono *diverse modalità di implementazione:***

- *Definizione di vincoli in linguaggio SQL, all'atto della definizione dello schema logico;*
- *Definizione di trigger;*
- *Definizione di opportune procedure in linguaggio di programmazione*

CAPITOLO 2: ELEMENTI DI TEORIA RELAZIONALE

2.1. Modello Relazionale

È basato sul concetto matematico di relazione

- **Dominio:** insieme di valori D ={ v1, … , vk }
- **Tupla:** dati n domini D1, .. Dn, non necessariamente distinti una tupla è definita come --> t = (v1, .. , vn) 1 <= i <= n
- **Prodotto cartesiano:** D1xD2x...xDn = insieme di tutte le tuple t su D1, … , Dn

- **Relazione:** una relazione si n domini non necessariamente distinti è un sottoinsieme del prodotto cartesiano D1x..xDn --> abbiamo che:

Valore di n = **grado** della relazione

Numero di tuple = **cardinalità** della relazione

- **Rappresentazione di relazioni:** tabella con un numero di righe pari alla cardinalità e numero di colonne pari al grado
- **Attributo:** nome dato ad un dominio di relazione

Si ottiene l'**indipendenza** dell'ordinamento dei domini

Si attribuisce **significato** ai valori del dominio

- **Schema di relazione:** è una coppia costituita dal nome della relazione R e da un insieme di nomi degli attributi X = (A1, .. , An) indicato con R(X)
- **Istanza di relazione:** su uno schema R(A1,., An) è insieme r di tuple su (A1, .. , An)
- **Schema di base di dati:** è un insieme di schemi di relazioni R = { R1(X1) , .. , Rn(Xn)}
- **Istanza di base di dati:** dato uno schema di basi di dati R = { R1(X1) , .. , Rn(Xn)}, una istanza su R è un insieme di relazioni

R = { r1, .. , rn} dove ri è una relazione su Ri, per ogni 1<=i<=n

Notazione:

- **Insieme di attributi:** un insieme di attributi Y dello schema R(X), cioè Y sottoinsieme di X, può essere denotato anche con R.Y
- **Tuple:** data una tupla t su R(X), un attributi A appartenente a X e un sottoinsieme Y di X --> si ha che:

t[A]= valore di t su A

t[Y]= sottotupla di t ottenuta considerando i valori degli attributi in Y

- **Chiave di relazione:** un sottoinsieme dei suoi attributi che identifica univocamente ogni tupla della relazione stessa

Dato uno schema di relazione R(X), un sottoinsieme di attributi K di X è detta ***chiave*** dello schema di relazione R(X) ***se e solo se*** per ogni relazione r su R(X) valgono le seguenti proprietà:

1) Univocità: non esistono due tuple distinte di r con lo stesso valore della chiave

2) Minimalità: non esiste un sottoinsieme proprio di K con la proprietà di univocità

--> se vale SOLO la proprietà di univocità: superchiave

Ogni schema di relazione R(X) ha ***almeno*** una chiave.

Notazione:

- **Chiave primaria:** si indica sottolineando gli attributi che la compongono

R(K1,K2, .., Km, A2, … , An)

- **Chiave alternativa:** è riportata di seguito allo schema e contraddistinta dalla parola chiave **AK:**

*R(X) --> **AK**: K1, .. , Km*

- **Valori nulli:** ogni dominio di relazione viene esteso con un particolare valore, detto valore nullo (NULL), che rappresenta assenza di informazione.

Esempio:

Schema di relazione DOCENTE (CODICE, CF, NOME, LUOGO_NASCITA)--> si possono elencare vari vasi in cui si deve inserire nella relazione una tupla il cui valore di un attributo non è disponibile.

- **Vincoli di integrità:** stabiliscono condizioni di correttezza delle informazioni nella base di dati.

A) Vincolo di Entity Integrity: vincolo che stabilisce che gli attributi che costituiscono la chiave primaria (o alternativa) di una relazione NON possono assumere valore nullo.

B) Foreign Key --> se specificata dichiara vincolo di integrità referenziale in uno schema di base di dati R. *Si definisce Foreign Key un insieme di attributi FK = {FK1,...FKn} di uno schema di relazione R1 appartenente ad R.*

C) Chiave della Relazione riferita: *schema di relazione R2 appartenente a R*, non necessariamente distinto da R1, *con una chiave K={K1,…,Km}, con m=n.*

Notazione:

R1(x)

FK: FK1, .. , FKn **REFERENCES** R2(K1, .., Kn)

2.2 Algebra relazionale

- Algebra relazionale è un insieme di operatori che si applicano alle relazioni e restituiscono relazioni.

- Tramite le *espressioni* dell'algebra relazionale è possibile formulare *interrogazioni* anche complesse sulla base di dati.

- L'algebra relazionale è composta da 5 operatori di base a partire dai quali si possono definire altri operatori, detti *derivati.*

Operatori Base:

1) **Unari:**
   - Selezione
   - Proiezione

2) **Binari:**
   - Unione
   - Differenza
   - Prodotto Cartesiano

3) **Principali operatori derivati:**
   - Intersezione
   - Join
   - Divisione

A) OPERATORI UNARI:

Selezione : Operatore σ

Data una relazione r con schema X e un predicato di selezione D

- F è un'espressione booleana di predicati semplici p
- Un predicato semplice p è il confronto tra due espressioni che utilizzano nomi di attributi costanti e operazioni aritmetiche

l'operazione di selezione σ_F (r) ha come risultato una relazione con schema X definita dal sottoinsieme di tuple di r che soddisfano il predicato F

$$\sigma_F (r) = \{t \mid t \in r, F(t) = true\}$$

Proiezione : Operatore π

Data una relazione r con schema X e un insieme di attributi $Y \subseteq X$, il risultato dell'operazione di proiezione π_Y (r) è una relazione con schema Y definita dall'insieme di sotto-tuple di r ottenute considerando solo i valori su Y:

$$\pi_Y (r) = \{t[Y] \mid t \in r\}$$

B) OPERATORI BINARI:

Unione : Operatore ∪

Date due relazioni r e s con lo stesso schema X, il risultato dell'operazione di unione $r \cup s$ è una relazione che ha come schema X ed è definita dall'unione delle tuple di r con le tuple di s

$$r \cup s = \{t \mid t \in r \vee t \in s\}$$

Differenza : Operatore −

Date due relazioni r e s con lo stesso schema X, l'operazione di differenza $r - s$ è una relazione che ha come schema X ed è definita dalla differenza tra le tuple di r e quelle di s

$$r - s = \{t \mid t \in r \wedge t \notin s\}$$

Prodotto Cartesiano : Operatore ×

Date due relazioni r e s con schema R(X) e S(Y), il risultato dell'operazione di prodotto cartesiano r × s è una relazione che ha come schema R.X ∪ S.Y ed è definita dall'insieme di tuple

$$r \times s = \{t \mid t = t_R t_S : t_R \in r \wedge t_S \in s\}$$

dove $t_R t_S$ indica la concatenazione della tupla t_R e della tupla t_S.

• Nello schema del prodotto cartesiano R.X ∪ S.Y , un attributo A che è solo in R.X (S.Y) può essere indicato semplicemente con A.

C) PRINCIPALI OPERATORI DERIVATI

Intersezione : Operatore ∩

L'operatore di intersezione ∩ è un operatore derivato definito come

$$r \cap s = r - (r - s) = \{t \mid t \in r \wedge t \notin (r - s)\} = \{t \mid t \in r \wedge t \in s\}$$

Theta-Join : Operatore 1_F

- L'operazione di join serve per combinare tuple prese da due o più.

  relazioni sulla base di condizioni espresse sugli attributi delle tuple.

- Il Theta-join tra r e s definisce un sottoinsieme del prodotto cartesiano

  r × s ottenuto tramite una operazione di selezione.

- Date due relazioni r e s con schema R(X) e S(Y), e un predicato di join F costituito da una espressione booleana di predicati di confronto AΘB, dove A ∈ X, B ∈ Y e Θ è un operatore di confronto, il Theta-join tra r e s è definito come

$$r 1_F s = \sigma_F(r \times s)$$

- **Equijoin**
  I predicati di confronto sono esclusivamente predicati di uguaglianza

Naturaljoin : Operatore 1

Date due relazioni r e s con schemi R(X) e S(Y), il naturaljoin (o join naturale) tra r e s, indicato con r 1 s, è il risultato dell'equijoin tra r e s su tutti gli attributi comuni a X e Y proiettato sull'unione degli attributi dei due schemi (XY)

$$r \,1\, s = \pi_{X \cup Y} (r \,1^{V_i}_{A_i \in X \cap Y} (R.A_i = S.A_i)\, s)$$

Semijoin : Operatore ⋉

Date due relazioni r e s con schemi R(X) e S(Y) il semijoin da s a r, indicato con r ⋉ s è la proiezione su X del join naturale di r e s

$$r \ltimes s = \pi_X (r \,1\, s)$$

Si può verificare che r ⋉ s = r 1 ($\pi_{X \cap Y}$ s)

OuterJoin :

- Il risultato di un join r 1 s comprende solo le tuple t_R di r (t_S di s) che possono essere messe in corrispondenza, in base al predicato di join, con una tupla t_S di s (t_R di r).

  Una tupla t_R di r (t_S di s) che non partecipa a tale corrispondenza, e che quindi non contribuisce al join, `e detta dangling.

  Più precisamente, una tupla t_R di r è detta dangling se non esiste t ∈ r⋉s tale che t[X]=t_R (stessa cosa per t_S di s).

- Gli operatori di OuterJoin servono per includere nel risultato del join anche le tuple dangling: tale tuple sono concatenate con tuple composte da valori nulli

Left- outerjoin: r =⟕ s
comprende le tuple dangling t_R di r

Right-outerjoin: r ⟖= s
comprende le tuple dangling t_S di s

full-outerjoin: r =⟗= s
comprende sia le tuple dangling t_R di r che le tuple dangling t_S di s

Divisione: operatore •/•

Date due relazioni r e s, l'operazione di divisione tra r (dividendo) e s (divisore), indicata r•/•s, serve per individuare le tuple di r associate a tutte le tuple di s. → siano R(X) e S(Y) gli schemi delle due relazioni tali che Y è contenuto in X, l'operazione di divisione tra r e s, ha come risultato una relazione che ha schema (X-Y) ed è definita da:

r•/•s={tD | per ogni tS appartenete ad s, tDtS appartenente a r}

CAPITOLO 3: PROGETTAZIONE LOGICA

3.1 Progetto logico di alto livello con il modello E/R

Fasi della progettazione logica:

1) ristrutturazione dello schema E/R

- È indipendente dal modello logico
- Si basa su criteri di ottimizzazione dello schema
- *Consiste nell'eliminazione di gerarchie e identificazioni esterne, normalizzazione di attributi composti o multipli, scelta di chiavi primarie.*

2) traduzione verso il modello logico:

- È riferita ad un particolare modello logico: **modello relazionale**
- Vengono effettuate *ulteriori ottimizzazioni* basate sulle caratteristiche del modello logico
- Consiste nella traduzione di entità e associazioni in schemi di relazioni

Carico di Lavoro:

Il carico di lavoro sul DB è rappresentato sia dalla dimensione dei dati che dalle operazioni più significative che si stima saranno eseguite dal DB.

Il 20% delle operazioni produce l'80% del carico.

- **Volume dei dati:** numero medio di istanze di ogni entità e associazione + cardinalità e dimensioni di ciascun attributi + percentuali di copertura gerarchiche
- **Tabella dei volumi: CONCETTO = nome | TIPO che può essere E, A, R(associazione) |VOLUME DEI DATI**
- **Descrizione delle operazioni:**

  Tipo dell'operazione: Interattiva o Batch

  Frequenza: numero medio di esecuzioni in un certo periodo di tempo

  Schema di operazione: frammento dello schema E/R interessato dall'operazione sul quale viene disegnato il "cammino logico" da percorrere per accedere alle informazioni di interesse

- **Tabella delle operazioni: OPERAZIONE | TIPO (I o B) | FREQUENZA**
- **Tabella degli accessi: CONCETTO | ACCESSI | TIPO**

OSS: il peso degli accessi in scrittura è il doppio di quello delle letture

Dati derivati:

Un dato derivato è un dato che può essere ottenuto attraverso una serie di operazioni da altri dati.

Sulla base delle operazioni e delle loro frequenze è possibile valutare se è conveniente o meno mantenere nello schema gli attributi derivati.

- **Vantaggio:** a tempo di accesso non è richiesta alcuna operazione per ricavare il valore dell'attributo
- **Svantaggi:** occorre eseguire operazioni di aggiornamento per mantenere la consistenza dei dati; si spreca memoria.

Attributi possono derivare da:

- Altri attributi della stessa entità o associazione
- Da attributi di altre entità o associazioni
- Operazioni di conteggio di istanze

Associazioni derivabili dalla composizione di altre associazioni

Eliminazione delle gerarchie:

Si ottiene uno schema E/R in cui ogni gerarchia è sostituita da appropriate entità e associazioni.

Le alternative che si presentano sono 3:

1) Mantenimento delle entità con associazioni: tutte le entità vengono mantenute --> le entità figlie sono in associazione binaria con l'entità padre e sono identificate esternamente

2) Collasso verso l'alto: *riunisce tutte le entità figlie nell'entità padre* --> soluzione che favorisce operazioni che consultano insieme gli attributi dell'entità padre e quelli di un'entità figlia: si accede ad una sola entità, anziché a due attraverso un'associazione ***+ bisogna inserire Selettore*** = attributo che specifica se una singola istanza di E appartiene ad una delle N sotto-entità. Esso di dipende dalla copertura.

3) Collasso verso il basso: *elimina l'entità padre trasferendone gli attributi e associazioni su tutte e due le entità figlie.* Un'associazione dell'entità padre si moltiplica, tante volte quante sono le figlie

- Se la **copertura non è totale** *non si può fare*
- Se la **copertura non è esclusiva** *introduce ridondanza*

--> soluzione interessante in presenza di molti attributi di specializzazione. Favorisce le operazioni in cui si accede separatamente a ciascuna delle entità figlie.

OSS:

1) le trasformazioni delle gerarchie possono modificare le cardinalità di attributi e associazioni;

2) l'applicabilità e convenienza delle soluzioni dipendono dalle proprietà di copertura della gerarchia e delle operazioni previste

3) le operazioni influiscono sulla convenienza delle varie soluzioni in base al modo in cui accedono a istanze della porzione di schema considerata.

Ulteriori trasformazioni:

- **Partizionamento di entità:** un'entità può essere partizionata orizzontalmente (istanze) o verticalmente (attributi) al fine di meglio rispondere alle operazioni previste --> può anche essere introdotto dalla presenza di diversi privilegi di accesso ed è *utile soprattutto nel progetto di DBMS distribuiti.*
- **Accorpamento di entità:** *due entità partecipanti ad un'**associazione uno-a-uno** possono essere accorpate in un'unica entità contenete gli attributi di entrambi.*

In associazione uno-a-molti l'accorpamento di entità genera ridondanza.

- **Partizionamento e accorpamento di associazioni:** partizionamento è tipicamente orizzontale.

3.2 Progettazione logica relazionale

3.2.1 trasformazione di attributi ed identificatori

Attributi multi-valore per le entità (pag. 101)

Se un'entità E ha un attributo multiplo A, si crea una nuova entità EA che ha A come attributo singolo ed è collegata ad E. Il tipo di collegamento dipende dai vari casi che verranno analizzati nel seguito, mostrando la semplificazione dell'attributo multi-valore e quindi la traduzione in relazionale:

- **Caso a)** *un valore può comparire una sola volta nella ripetizione*
- **Caso b)** *un valore può comparire più volte nella ripetizione*
- **Caso c)** *cardinalità massima nota: max-card(A,E)=K*

Attributi multi-valore per le associazioni

Conviene reificare l'associazione: si ha un attributo multiplo su un'entità e si applicano le regole già viste.

Risoluzione degli identificatori

Ogni **identificatore interno** dell'entità E corrisponde ad una chiave candidata della relazione E. --> **scelta della chiave primaria:** *è necessario che tra le diverse chiavi candidate di un'entità sia designata una chiave primaria.*

Criteri Euristici:

- Scegliere la chiave che è usata più frequentemente per accedere direttamente alle istanze dell'entità
- Preferire chiavi semplici a chiavi composte
- Eventualmente introdurre una chiave surrogata, ovvero un attributo che non ha significato e di cui è garantita l'unicità all'interno dell'entità (numero progressivo, appositi codici interni di identificazione).

Eliminazione degli identificatori esterni

È un'operazione di semplificazione già riferita nel modello logico.

REGOLA: Una componente di identificazione esterna di un'entità E2 da un'entità E1 tramite un'associazione R comporta il trasporto dell'identificatore di E1 su E2. Nel modello relazionale, l'identificatore di E1 trasportato nella relazione E2 diventa:

- *Una parte della chiave di E2*
- *Una foreign key riferita alla relazione E1 –>* l'associazione R tra E1 ed E2 è automaticamente tradotta e può essere eliminata.

3.2.2 Traduzione di entità e associazioni

Traduzione standard

- Ogni entità è tradotta con una relazione con gli stessi attributi:

--> la chiave primaria della relazione è quella dell'entità stessa

--> ogni attributo identificatore dell'entità è chiave alternativa della relazione

- Ogni associazione R tra le entità E1, .. , En, è tradotta con una relazione con gli stessi attributi, cui si aggiungono le chiavi primarie di tutte le entità che essa collega:

--> ogni chiave primaria di un'entità Ei tale che max-card(Ei,R)=1, è una chiave di relazione R; altrimenti la chiave della relazione è composta dall'insieme di tutte le chiavi primarie delle entità collegate

--> le chiavi primarie dell'entità collegate sono chiavi straniere NOT NULL (FK) riferite alle corrispondenti entità

--> se la chiave straniera fa parte di una delle chiavi non è necessario indicare NOT NULL

OSS: traduzione standard

- **Traduzione che è sempre possibile**
- **È l'unica possibilità per le associazioni in cui tutte le entità partecipano con molteplicità maggiore di uno**

1) Associazione binaria uno a uno tradotta con una relazione:

- Se ***l'associazione è obbligatoria per entrambe*** le entità la chiave primaria può essere indifferentemente K1 o K2 (altra è alternativa); ogni identificatore delle entità è chiave della relazione
- Se ***l'associazione è parziale per un'entità*** (E1) ***e obbligatoria per l'altra*** (E2), la chiave deve essere K1; saranno possibili valori nulli per gli attributi di E2 e di R
- Se ***l'associazione è parziale per entrambe***, occorre che entrambe le entità abbiano lo stesso identificatore; la chiave può essere indifferentemente K1 o K2; saranno possibili valori nulli per gli attributi di E1, E2 e R.

2) Associazione binaria uno a uno tradotta con due relazioni:

- L'associazione si può compattare in una delle entità, diciamo E1, includendo in E1 gli attributi di R e la chiave primaria di E2 come foreign key.
- Per evitare i valori nulli, è preferibile compattare l'associazione in un'entità che partecipa obbligatoriamente.

3) Associazione binaria uno a uno tradotta con tre relazioni

- È preferibile se l'associazione è parziale per entrambe le entità. Praticamente forzata se, oltre alla parzialità, le due chiavi primarie delle entità hanno domini distinti.

4) Associazioni binarie uno a molti

- **Traduzione con due relazioni**

L'associazione può essere compattata nell'entità che partecipa con molteplicità unaria, diciamo E1, includendo in E1 gli attributi di R e la chiave primaria di E2 come foreign key:

E2(K2, B1, B2...)

E1(K1, A1,..,K2,C1,C2..) **FK:** K2 **REFERENCES** E2

- **Traduzione con tre relazioni**

Se la partecipazione di E1 è parziale, per evitare i valori nulli, si può optare per la traduzione standard con 3 relazioni

5) Associazioni unarie --> può dare luogo ad una o due relazioni, dipendentemente dalle molteplicità in gioco

- **Anello molti a molti**

È tradotto con due relazioni, una per l'entità e una per l'associazione; la chiave della relazione che modella l'associazione è composta da due attributi, i cui nomi riflettono il diverso ruolo dell'entità; ognuno di questi due attributi è anche foreign key

- **Anello uno a molti**

Oltre che con due relazioni, è traducibile con una sola relazione che contiene due volte l'attributo identificatore: una volta come chiave primaria e una volta come chiave esterna con un nome che riflette il ruolo dell'entità

- **Anello uno a uno (pag.110)**

6) Associazione n-aria

È normalmente tradotta seguendo la traduzione standard

- In presenza di entità Ei che partecipano all'associazione n-aria R con max-card(Ei,R)=1, sono possibili altre traduzioni applicabili in casi particolari

CAPITOLO 5: TEORIA DELLA NORMALIZZAZIONE

5.1. Dipendenze funzionali

Uno degli elementi di conoscenza disposizione del progettista che deve modellare un'applicazione database è che esistono delle "dipendenze tra dati". In maniera astratta, questo viene modellato dal concetto di "dipendenza funzionale"

- **Convenzioni di notazioni:**
  1) *Prime lettere* dell'alfabeto maiuscole (A,B,C,D,E) denotano *un attributo*
  2) *Le ultime lettere* dell'alfabeto maiuscole (U,V,X,Y,Z) denotano *insieme di attributi*
  3) R denota lo schema di una relazione, r l'istanza corrente dello schema R
  4) Dato un insieme di attributi A1,..,An, uno schema di relazione R avente tale insieme di attributi verrà denotato indifferentemente con R(A1,..,An) oppure con R(A1A2…An).

- **Dipendenza funzionale:** *dato uno schema di relazione R(X) una dipendenza funzionale (FD) su R è un vincolo di integrità espresso nella forma Y→Z, dove Y e Z sono sottoinsiemi di X; in tal caso si dice che Y determina funzionalmente Z.*

Un'istanza r di R soddisfa la dipendenza funzionale Y→Z se in ogni tupla di r dato il valore di Y determina univocamente il valore di Z.

È facile verificare che se Y sottoinsieme di Z allora Y→Z. queste dipendenze funzionali sono dette **banali.**

- **Istanza legale:** dato uno schema <R(T),F>, un'istanza r di R(T) è detta istanza legale di <R(T),F> se soddisfa tutte le dipendenze funzionali in F. Uno schema <R(T),F> dà luogo a istanze legali che oltre alle FD in F soddisfano necessariamente altre FD.

- **Implicazione Logica:** dato uno schema <R(T),F> e una dipendenza funzionale X→Y, si dice che F implica logicamente X→Y, denotata con F=X→Y, se ogni istanza legale r di <R(T),F> soddisfa anche X→Y.
- **Chiusura di un insieme di FD:** dato uno schema <R(T),F>, la chiusura di F, denotato con F+, è l'insieme delle dipendenze funzionali implicate logicamente da F:
  F+ ={X→Y | F=X→Y}

- **Chiave:** dato <R(T),F>, un insieme K contenuti in T è detto chiave di <R(T),F> se:
  a) K→T è in F+
  b) Non esiste nessun sottoinsieme proprio Y di K tale che Y→T è in F+

→ con il termine **super-chiave** denotiamo un qualsiasi sopra-insieme di una chiave

5.3 Forme normali

Sono alla base della teoria della normalizzazione di schemi relazionali, il cui *obiettivo* principale è quello di *definire formalmente la qualità degli schemi*.

La qualità di uno schema viene definita come ***l'assenza*** di:

- **Ridondanza nei dati**
- **Anomalie di aggiornamento dei dati**

1) **Prima forma normale - 1NF:** nasce dall'esigenza di introdurre un modello dei dati particolarmente semplice, piatto, che sta alla base di semplici linguaggi di interrogazione e manipolazione.
   → **si definisce 1NF:** *uno schema R(X) è in 1NF se e solo se i valori di tutti i domini degli attributi A appartenenti a X sono atomici.*

- Normalmente nel modello relazionale, si assume come implicito il vincolo di avere domini atomici per gli attributi.
- In altri modelli il vincolo di avere domini atomici è rimosso e un attributo può avere come valore, oltre ad un valore elementare, anche un valore complesso, quale ad esempio una tupla, un insieme oppure una loro combinazione.

2) **Seconda forma normale - 2NF:** *impedisce anomalie di aggiornamento dei dati e diminuisce la ridondanza***.**
   + Dato uno schema <R(T),F>, un attributo A appartenente a T e un insieme di attributi Y contenuto in T, *si diche che A dipende completamente* da Y se Y→A appartenente a F e non esiste nessun sotto-insieme proprio Z di Y, Z contenuto in Y, tale che Z→A.
   → **si definisce 2NF:** *uno schema <R(T),F> è in 2NF **se e solo se** ogni attributo **non** primo A appartenete a T **dipende completamente** da **ognuna** delle chiavi di R.*
   Oss: F contiene solo dipendenze funzionali del tipo X → A

- **Attributo primo:** un attributo si dice primo se e solo se appartiene ad una chiave → un attributo che non appartiene ad alcuna chiave è non primo.

3) **Terza forma normale – 3NF:** serve ad evitare problemi derivanti da attributi non super-chiave che determinano funzionalmente altri attributi non primi.
   → **si definisce 3NF:** *uno schema <R(T),F> è in 3NF **se e solo se** per ogni dipendenza funzionale non banale X→A appartenete a F, o X è una super-chiave oppure A è primo.*

   ***+ oss:** uno schema <R(T),F> che è in 3NF è anche in 2NF.*

   La 3NF non è esente da problemi di ridondanza e anomalie di aggiornamento dei dati → definisce comunque schemi con un buon livello di qualità e costituisce lo standard da noi adottato per il progetto logico.

4) **Forma normale di Boyce-Codd (BCNF):** afferma che tutti gli attributi (anche quelli primi) dipendono funzionalmente solo dalle super-chiavi.
   → **si definisce BCNF:** *uno schema <R(T),F> è in BCNF **se e solo se** per ogni dipendenza funzionale non banale X→A appartenete a F, X è una super-chiave.*

   ***+ oss:** uno schema <R(T),F> che è in BCNF è anche in 3NF.*