UNA TABELLA

N Inv	Stanza	Resp	Oggetto	Produttore	Descrizione
1012	256	Ghelli	Mac Mini	Apple	Personal Comp
1015	312	Albano	Dell XPS M1330	Dell	Notebook 2 GHZ
1034	256	Ghelli	Dell XPS M1330	Dell	Notebook 2GB
1112	288	Leoni	Mac Mini 2	Apple	Personal Comp

È fatta male? Perché? Come si può correggere?

TEORIA RELAZIONALE: INTRODUZIONE

- Due metodi per produrre uno schema relazionale:
  a) Partire da un buon schema a oggetti e tradurlo
  b) Partire da uno schema relazionale fatto da altri e modificarlo o completarlo
- Teoria della progettazione relazionale: studia cosa sono le "anomalie" e come eliminarle.
- È particolarmente utile se si usa il metodo (b). È moderatamente utile anche quando si usa il metodo (a).

SCHEMI CON ANOMALIE

- Esempio:
  - StudentiEdEsami(Matricola, Nome, Provincia,AnnoNascita, Materia, Voto)
- Anomalie:
  - Ridondanze
  - Potenziali inconsistenze
  - Anomalie nelle inserzioni
  - Anomalie nelle eliminazioni
- Schema senza anomalie
  - Studenti (Matricola, Nome, Provincia, AnnoNascita)
  - Esami (Materia, Matricola, Voto)

OBIETTIVI

- Nozione base: dipendenze funzionali
- Obiettivi della teoria:
  - Equivalenza di schemi
  - Qualità degli schemi (forme normali)
  - Trasformazione degli schemi (normalizzazione di schemi)
- Ipotesi dello schema di relazione universale:
  - Tutti i fatti sono descritti da attributi di un'unica relazione (relazione universale), cioè gli attributi hanno un significato globale.

DIPENDENZE FUNZIONALI

- Per formalizzare la nozione di schema senza anomalie, occorre una descrizione formale della semantica dei fatti rappresentati in uno schema relazionale.
- Istanza valida di R: è una nozione semantica, che dipende da ciò che sappiamo del dominio del discorso

DIPENDENZE FUNZIONALI

- Dato uno schema $R(T)$ e $X, Y \subseteq T$, una dipendenza funzionale (DF) è un vincolo su R del tipo $X \rightarrow Y$, i.e. X determina funzionalmene Y o Y è determinato da X, se per ogni istanza valida di R un valore di X determina in modo univoco un valore di Y:

  $\forall$ r istanza valida di R.

  $\forall$ t1, t2 $\in$ r. se t1[X] = t2[X] allora t1[Y] = t2[Y]

- Si dice che un'istanza r_0 di R *soddisfa* le DF $X \rightarrow Y$ ($r_0 \models X \rightarrow Y$) se la proprietà vale per r_0, e che un'istanza r_0 di R *soddisfa* un insieme F di DF se, per ogni $X \rightarrow Y \in F$, vale $r_0 \models X \rightarrow Y$:
  - $r_0 \models X \rightarrow Y$ *sse* $\forall$ t1, t2 $\in r_0$. se t1[X] = t2[X] allora t1[Y] = t2[Y]

ESEMPIO

- DotazioniLibri(CodiceLibro, NomeNegozio, IndNegozio, Titolo, Quantità)
- DF:

{ CodiceLibro $\rightarrow$ Titolo

NomeNegozio $\rightarrow$ IndNegozio

CodiceLibro, NomeNegozio $\rightarrow$ IndNegozio, Titolo, Quantità }

ESPRIMERE LE DIPENDENZE FUNZIONALI

- Consideriamo: NomeNegozio $\rightarrow$ IndNegozio
- Espressione diretta:
  - Se in due righe il NomeNegozio e uguale, anche l'IndNegozio è uguale:
    - $\text{NomeNegozio}_{=} \Rightarrow \text{IndNegozio}_{=}$
- Per contrapposizione:
  - Se l'IndNegozio è diverso allora il NomeNegozio è diverso:
    - $\text{IndNegozio}_{\neq} \Rightarrow \text{NomeNegozio}_{\neq}$
- Per assurdo:
  - Non possono esserci due dotazioni con NomeNegozio uguale e IndNegozio diverso:
    - $\text{Not } (\text{NomeNegozio}_{=} \wedge \text{IndNegozio}_{\neq})$
    - $\text{NomeNegozio}_{=} \wedge \text{IndNegozio}_{\neq} \Rightarrow \text{False}$

MANIPOLAZIONE DI CLAUSOLE

- Sono equivalenti:
    - $NomeNegozio_{=} \Rightarrow IndNegozio_{=}$
    - $IndNegozio_{\neq} \Rightarrow NomeNegozio_{\neq}$
    - $NomeNegozio_{=} \wedge IndNegozio_{\neq} \Rightarrow False$
- In generale:
    - $A \Rightarrow B \quad \Leftrightarrow \quad A \wedge \neg B \Rightarrow False \quad \Leftrightarrow \quad \neg B \Rightarrow \neg A$
- Più in generale, in ogni clausola $A \wedge B \Rightarrow E \vee F$ posso spostare le sottoformule da un lato all'altro, negandole
- Quindi sono equivalenti:
    - $NomeNegozio_{=} \wedge CodiceLibro_{=} \Rightarrow Quantità_{=}$
    - $NomeNegozio_{=} \wedge CodiceLibro_{=} \wedge Quantità_{\neq} \Rightarrow False$
    - $CodiceLibro_{=} \wedge Quantità_{\neq} \Rightarrow NomeNegozio_{\neq}$
    - $NomeNegozio_{=} \wedge Quantità_{\neq} \Rightarrow CodiceLibro_{\neq}$

ESEMPIO

- Orari(CodAula, NomeAula, Piano, Posti, Materia, CDL, Docente, Giorno, OraInizio, OraFine)
- In un dato momento, un docente si trova al più in un'aula
- Non è possibile che due docenti diversi siano nella stessa aula contemporaneamente
- Se due lezioni si svolgono su due piani diversi appartengono a due corsi di laurea diversi
- Se due lezioni *diverse* si svolgono lo stesso giorno per la stessa materia, appartengolo a due CDL diversi (lezioni diverse: not($CodAula_{=}$ $\wedge$ and $NomeAula_{=}$ $\wedge$ ...))

DIPENDENZE FUNZIONALI

- Notazione:
  - R <T, F> denota uno schema con attributi T e dipendenze funzionali F.
- *Le DF sono una proprietà semantica, cioè dipendono dai fatti rappresentati e non da come gli attributi sono combinati in schemi di relazione.*
- Si parla di DF complete quando $X \rightarrow Y$ e per ogni $W \subset X$, non vale $W \rightarrow Y$.
- Se X è una superchiave, allora X determina ogni altro attributo della relazione: $X \rightarrow T$
- Se X è una chiave, allora $X \rightarrow T$ è una DF completa

PROPRIETÀ DELLE DF

- Da un insieme F di DF, in generale altre DF sono 'implicate' da F.
- *Definizione*: Sia F un insieme di DF sullo schema R, diremo che F implica logicamente $X \rightarrow Y$ ($F \models X \rightarrow Y$,), se ogni istanza r di R che soddisfa F soddisfa anche $X \rightarrow Y$.

ESEMPIO

- Sia r un'istanza di R<T, F>, con $F = \{X \rightarrow Y, X \rightarrow Z\}$ e $X, Y, Z \subseteq T$. Sia $X' \subseteq X$. Altre DF sono soddisfatte da r, ad es.
  - $X \rightarrow X'$ (DF *banale*) e
  - $X \rightarrow YZ$, infatti

    $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$

    $t_1[X] = t_2[X] \Rightarrow t_1[Z] = t_2[Z]$

    $t_1[X] = t_2[X] \Rightarrow t_1[YZ] = t_2[YZ]$

  Pertanto $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$

- Altro esempio: $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$

REGOLE DI INFERENZA

- Come derivare DF implicate logicamente da F, usando un insieme di regole di inferenza.
- “Assiomi” (sono in realtà regole di inferenza) di Armstrong:
  - Se $Y \subseteq X$, allora $X \rightarrow Y$ (Riflessività R)
  - Se $X \rightarrow Y$, $Z \subseteq T$, allora $XZ \rightarrow YZ$ (Arricchimento A)
  - Se $X \rightarrow Y$, $Y \rightarrow Z$, allora $X \rightarrow Z$ (Transitività T)

DERIVAZIONE

- **Definizione** Sia F un insieme di DF, diremo che $X \rightarrow Y$ sia *derivabile* da F ($F \vdash X \rightarrow Y$), *sse* $X \rightarrow Y$ può essere inferito da F usando gli assiomi di Armstrong.
- Si dimostra che valgono anche le regole:
  - $\{X \rightarrow Y, X \rightarrow Z\} \vdash X \rightarrow YZ$ (unione **U**)
  - $Z \subseteq Y$ $\{X \rightarrow Y\} \vdash X \rightarrow Z$ (decomposizione **D**)
- Da **U** e **D** si ricava che se $Y = A_1 A_2 \ldots A_n$ allora
  - $X \rightarrow Y \Leftrightarrow \{X \rightarrow A_1, X \rightarrow A_2, \ldots, X \rightarrow A_n\}$

ESEMPIO

- R(A B C D)
- $F = \{A \rightarrow B, BC \rightarrow D\}$
- AC è una superchiave? Ovvero $AC \rightarrow ABCD$?

1. $A \rightarrow B$	ipotesi 1
2. $AC \rightarrow BC$	da 1 per **Arr** (C)
3. $BC \rightarrow D$	ipotesi 2
4. $BC \rightarrow BCD$	da 3 per **Arr** (BC)
5. $AC \rightarrow BCD$	da 2+4 per **Trans**
6. $AC \rightarrow ABCD$	da 5 per **Arr** (A)

CORRETTEZZA E COMPLETEZZA DEGLI ASSIOMI DI ARMSTRONG

- **Teorema**. Gli assiomi di Armstrong sono corretti e completi.
- Correttezza degli assiomi:

$$\forall f, \qquad F \mid\!- f \;\Rightarrow\; F \models f$$

- Completezza degli assiomi:

$$\forall f, \qquad F \models f \;\Rightarrow\; F \mid\!- f$$

CHIUSURA DI UN INSIEME F

- **Definizione** Dato un insieme F di DF, la chiusura di F, denotata con F^+, è:

  $F^+ = \{ X \rightarrow Y \mid F \mid\!- X \rightarrow Y\}$

- **Definizione** Dato R<T, F>, e $X \subseteq T$, la *chiusura* di X rispetto ad F, denotata con X_F^+, (o X^+, se F è chiaro dal contesto) è

  $X_F^+ = \{A_i \in T \mid F \mid\!- X \rightarrow A_i\}$.

- **Problema dell'implicazione**: controllare se una DF $V \rightarrow W \in F^+$

  Un algoritmo efficiente per risolvere il problema dell'implicazione senza calcolare la chiusura di F scaturisce dal seguente teorema.

  Teorema $F \mid\!- X \rightarrow Y \Leftrightarrow Y \subseteq X_F^+$.

CHIUSURA LENTA

- Un semplice algoritmo per calcolare X^+ (ne esiste uno migliore di complessità di tempo $O(ap)$) è
- **Algoritmo** CHIUSURA LENTA

  input $R\langle T, F\rangle$ $X \subseteq T$

  output X^+

  begin

  $X^+ = X$

  while (X^+ cambia) **do**

  for $W \rightarrow V$ **in** F **with** $W \subseteq X^+$ **and** $V \not\subseteq X^+$

  do $X^+ = X^+ \cup V$

  end

ESEMPIO

- $F = \{DB \rightarrow E, B \rightarrow C, A \rightarrow B\}$, trovare $(AD)^+$:

$X^+ = AD$

$X^+ = ADB$

$X^+ = ADBE$

$X^+ = ADBEC$

CHIAVI E ATTRIBUTI PRIMI

- **Definizione** Dato lo schema R<T, F>, diremo che $W \subseteq T$ è una chiave candidata di R se

  1. $W \rightarrow T \in F^+$ (W superchiave)

     $\forall\ V \subset W, V \rightarrow T \notin F^+$ (se $V \subset W$, V non superchiave)

- *Attributo primo* : attributo che appartiene ad almeno una chiave
- Complessità
  - Il problema di trovare tutte le chiavi di una relazione richiede un algoritmo di complessità esponenziale nel caso peggiore
  - Il problema di controllare se un attributo è primo è NP-completo

TROVARE TUTTE LE CHIAVI

- Sia $F = \{C \rightarrow D, CF \rightarrow B, D \rightarrow C, F \rightarrow E\}$
- Ogni chiave deve contenere AF; le chiavi sono in $AF \cdot \mathcal{P}(BCDE) = AF^{BCDE}$ (nel testo: AF::(BCDE))
- $AF^{+} = AFE$; ogni chiave in $AF^{BCD} - \{AF\}$
- Candidati: $AF^{BCD} - \{AF\} = AFB^{CD} + AFC^{D} + AFD$

AF — B — ~~C~~, ~~D~~ ; AF — C ; AF — D

$AF^{+} = AFE$ $BCDE-AFE = BCD$

$AFB^{+} = AFBE$ $CD-AFBE = CD$

~~C~~ no: AFC chiave

~~D~~ no: AFD chiave

$AFC^{+} = AFCDBE$ **AFC chiave**

$AFD^{+} = AFDCEB$ **AFD chiave**

COPERTURA DI INSIEMI DI DF

- **Definizione:** Due insiemi di DF, F e G, sullo schema R sono equivalenti, $F \equiv G$, sse $F^+ = G^+$. Se $F \equiv G$, allora F è una copertura di G (e G una copertura di F).

- **Definizione** Sia F un insieme di DF:

1. Data una $X \to Y \in F$, si dice che X contiene un attributo *estraneo* A_i sse $(X - \{A_i\}) \to Y \in F^+$, cioè $F \mid\!- (X - \{A_i\}) \to Y$
2. $X \to Y$ è una dipendenza *ridondante* sse

   $(F - \{X \to Y\})^+ = F^+$, cioè $F - \{X \to Y\} \mid\!- X \to Y$

F è detta una copertura *canonica* sse

- la parte destra di ogni DF in F è un attributo;
- non esistono attributi estranei;
- nessuna dipendenza in F è ridondante.

ESISTENZA DELLLA COPERTURA CANONICA

- *Teorema* Per ogni insieme di dipendenze F esiste una copertura canonica.
- Algoritmo per calcolare una copertura canonica:
  - Trasformare le dipendenze nella forma $X \rightarrow A$
  - Eliminare gli attributi estranei
  - Eliminare le dipendenze ridondanti

DECOMPOSIZIONE DI SCHEMI

- In generale, per eliminare anomalie da uno schema occorre decomporlo in schemi più piccoli "equivalenti"
- **Definizione** Dato uno schema $R(T)$,
  $\rho = \{R_1(T_1), \ldots, R_k(T_k)\}$ è una decomposizione di R *sse* $\cup T_i = T$:
  - {Studenti(Matr,Nome), Esami(Matr,Materia)}
    decomp. di Esami(Matr,Nome,Materia)
  - {Studenti(Matr,Nome), Esami(Materia)}
  - {Studenti(Matr,Nome), Esami(Nome,Materia)}
- Due proprietà desiderabili di una decomposizione:
  - conservazione dei dati (*nozione semantica*)
  - conservazione delle dipendenze

DECOMPOSIZIONE DI SCHEMI

- Decomposizioni che preservano i dati:
- *Definizione*: $\rho = \{R_1(T_1), \ldots, R_k(T_k)\}$ è una decomposizione di $R(T)$ che preserva i dati sse per ogni *istanza valida* r di R:

  $$r = (\pi_{T1}\, r) \bowtie (\pi_{T2}\, r) \bowtie \ldots \bowtie (\pi_{Tk}\, r)$$

- Dalla definizione di giunzione naturale scaturisce il seguente risultato:
- *Teorema*: Se $\rho = \{R_1(T_1), \ldots, R_k(T_k)\}$ è una decomposizione di $R(T)$, allora per ogni istanza r di R:

  $$r \subseteq (\pi_{T1}\, r) \bowtie (\pi_{T2}\, r) \bowtie \ldots \bowtie (\pi_{Tk}\, r)$$

ESEMPIO DI DECOMPOSIZIONE

- Sia r qui sotto un'istanza valida di R(ABC):

r=

A	B	C
a1	b	c1
a2	b	c2

Allora la decomposizione {R(AB), R(BC)}:

π_{T1} r =

A	B
a1	b
a2	b

π_{T2} r =

B	C
b	c1
b	c2

non preserva i dati, infatti $\pi_{T1}\, r \bowtie \pi_{T2}\, r \supseteq r$

DECOMPOSIZIONI BINARIE

- **Teorema** Sia $R<T, F>$ uno schema di relazione, la decomposizione $\rho = \{R_1(T_1), R_2(T_2)\}$ preserva i dati sse
  - $T_1 \cap T_2 \rightarrow T_1 \in F^+$ oppure $T_1 \cap T_2 \rightarrow T_2 \in F^+$.

- Esistono criteri anche per decomposizioni in più di due schemi.

PROIEZIONE DELLE DIPENDENZE

- **Definizione** Dato lo schema R<T, F>, e $T_1 \subseteq T$, la proiezione di F su T_1 è

  $\pi_{T1}(F) = \{X \to Y \in F^+ \mid X Y \subseteq T_1\}$

- **Esempio**

  Sia R(A, B, C) e $F=\{A\to B, B\to C, C\to A\}$.

  $\pi_{AB}(F) \equiv \{A \to B, B \to A\}$

  $\pi_{AC}(F) \equiv \{A \to C, C \to A\}$

- Algoritmo banale per il calcolo di $\pi_{T1}(F)$:

  for each $Y \subseteq T_1$ **do** ($Z := Y^+$; output $Y \to Z \cap T_1$)

PRESERVAZIONE DELLE DIPENDENZE

- **Definizione** Dato lo schema $R<T, F>$, la decomposizione $\rho = \{R_1, ..., R_n\}$ preserva le dipendenze sse l'unione delle dipendenze in $\pi_{T_i}(F)$ è una copertura di F.
- **Proposizione** Dato lo schema $R<T, F>$, il problema di stabilire se la decomposizione $\rho = \{R_1, ..., R_n\}$ preserva le dipendenze ha complessità di tempo polinomiale.
- Un teorema importante:

  Teorema Sia $\rho = \{R_i<T_i, F_i>\}$ una decomposizione di $R<T, F>$ che preservi le dipendenze e tale che un T_j sia una superchiave per R. Allora ρ preserva i dati.

ESEMPIO

- Telefoni(Prefisso, Numero, Località, Abbonato, Via) $\{P\ N \rightarrow L\ A\ V,$ $L \rightarrow P\}$
- Si consideri la decomposizione:

  $\rho = \{Tel\langle\{N, L, A, V\}, F1\rangle, Pref\langle\{L, P\}, F2\rangle\}$ con

  - $F1 = \{LN \rightarrow A\ V\}$
  - $F2 = \{L \rightarrow P\}$
- Preserva dati ma non le dipendenze: $PN \rightarrow L$ non è deducibile da F1 e F2.

FORME NORMALI

- **1FN**: Impone una restrizione sul tipo di una relazione: ogni attributo ha un tipo elementare.
- **2FN**, **3FN** e **FNBC**: Impongono restrizioni sulle dipendenze. FNBC è la più naturale e la più restrittiva.
- **FNBC**:
  - Intuizione: se esiste in R una dipendenza $X \rightarrow A$ non banale ed X non è chiave, allora X modella l'identità di un'entità diversa da quelle modellate dall'intera R
  - Ad esempio, in StudentiEdEsami, il Nome dipende dalla Matricola che *non* è chiave.

FNBC

- **Definizione** R<T, F> è in BCNF $\Leftrightarrow$ per ogni $X \rightarrow A \in F^+$, con $A \notin X$ (non banale), X è una superchiave.
- **Teorema** R<T, F> è in BCNF $\Leftrightarrow$ per ogni $X \rightarrow A \in F$ non banale, X è una superchiave.
- **Esempi**:
  - Docenti(<u>CodiceFiscale</u>, Nome, Dipartimento, Indirizzo)
  - Impiegati(<u>Codice</u>, Qualifica, <u>NomeFiglio</u>)
  - Librerie(CodiceLibro, NomeNegozio, IndNegozio, Titolo, Quantità)
  - Telefoni(Prefisso, Numero, Località, Abbonato, Via)
    - $F = \{P\ N \rightarrow L\ A\ V,\ L \rightarrow P\}$

L'ALGORITMO DI ANALISI

- R<T,F> è decomposta in: $R_1(X, Y)$ e $R_2(X, Z)$ e su di esse si ripete il procedimento; esponenziale.

ρ= {R<T, F>}

while esiste in ρ una R_i<T_i, F_i> non in BCNF per la DF $X \rightarrow A$

do

$T_a = X^+$

$F_a = \pi_{Ta}(F_i)$

$T_b = T_i - X^+ + X$ $\leftarrow$ *attenzione: errore nel vecchio libro*

$F_b = \pi_{Tb}(F_i)$

$\rho = \rho - R_i + \{R_a\langle T_a, F_a\rangle, R_b\langle T_b, F_b\rangle\}$

(R_a ed R_b sono nomi nuovi)

end

PROPRIETA' DELL'ALGORITMO

- Preserva i dati, ma non necessariamente le dipendenze
- Esempi di decomposizioni senza perdita di dipendenze:
  - Docenti(CodiceFiscale, Nome, Dipartimento, Indirizzo), {CF $\rightarrow$ N D; D $\rightarrow$ I}
    - R1(D,I); R2(CF,N,D)
  - Impiegati(Codice, Qualifica, NomeFiglio) {C $\rightarrow$ Q}
    - R1(C, Q); R2(C, NF)

PROPRIETA' DELL'ALGORITMO (cont.)

- Telefoni(<u>Prefisso</u>, <u>Numero</u>, Località, Abbonato, Via),
  $\{P\ N \rightarrow L\ A\ V,\ L \rightarrow P\}$
  - R1(<u>L</u>, P); R2(<u>L</u>, <u>N</u>, A, V)
  - Preserva dati ma non le dipendenze: PN $\rightarrow$ L non è deducibile da F1 e F2.
- Cosa vuole dire "non preserva le dipendenze"?
  - R1 = {(Pisa, 050); (Calci, 050)}
  - R2 = {(Pisa, 506070, Rossi, Piave),
    (Calci,506070, Bianchi, Isonzo)}

TERZA FORMA NORMALE

- **Definizione:** R<T, F> è in 3FN se per ogni $X \rightarrow A \in F^{+}$, con $A \notin X$, X è una superchiave o A è primo.
- La 3FN ammette una dipendenza non banale e non-da-chiave se gli attributi a destra sono primi; la BCNF non ammette mai nessuna dipendenza non banale e non-da-chiave.
- **Teorema:** R<T, F> è in 3FN se per ogni $X \rightarrow A \in F$ non banale, allora X è una superchiave oppure A è primo.

ESEMPI

- Non sono in 3FN (e quindi, neppure in BCNF)
  - Docenti(<u>CodiceFiscale</u>, Nome, Dipartimento, Indirizzo)
  - Impiegati(<u>Codice</u>, Qualifica, <u>NomeFiglio</u>)
- Sono in 3FN, ma non in BCNF:
  - Telefoni(Prefisso, Numero, Località, Abbonato, Via)
    - $F = \{P\ N \rightarrow L\ A\ V,\ L \rightarrow P\}$
    - $K = \{PN, LN\}$
  - Esami(Matricola, Telefono, Materia, Voto)
    - Matricola Materia → Voto
    - Matricola → Telefono
    - Telefono → Matricola
    - Chiavi: Matricola Materia, Telefono Materia

L'ALGORITMO DI SINTESI: VERSIONE BASE

- Sia R<T, F>, con F **copertura canonica** e tutti gli attributi interessati da qualche DF.
  1. Si partiziona F in gruppi tali che ogni gruppo ha lo stesso determinante.
  2. Si definisce uno schema di relazione per ogni gruppo, con attributi gli attributi che appaiono nelle DF del gruppo, e chiavi i determinanti.
  3. Si eliminano schemi contenuti in altri.
  4. Se la decomposizione non contiene uno schema i cui attributi sono una superchiave di R, si aggiunge lo schema con attributi W, con W una chiave di R.

LE DF NON BASTANO: DIPENDENZE MULTIVALORE

- Impiegati(Codice, StoriaStipendio, NomeFiglio)

c1	s1	n1
c1	s1	n2
c1	s2	n1
c1	s2	n2

- La coesistenza di due proprietà multivalore INDIPENDENTI, fa sì che per ogni impiegato esistono tante ennuple quante sono le possibili coppie di valori di Qualifica e NomeFiglio.

Impiegati
Codice Qualifiche: seq num NomeFigli: seq string

Impiegati
Codice Posizioni: seq (Qualifica, NomeDirigente)