Attributi e domini

- Assumiamo un universo infinito numerabile $\mathcal{U} = \{A_0, A_1, A_2 \ldots\}$ di *attributi*.
- Denotiamo gli attributi con $A, B, C, B_1, C_1 \ldots$ e gli insiemi di attributi con $X, Y, Z, X_1, \ldots$
- per brevità scriviamo: A per $\{A\}$; XY per $X \cup Y$ (pertanto $A_1A_2A_3$ denota $\{A_1, A_2, A_3\}$).
- Ad ogni attributo A è associato un insieme di *valori* $\Delta[A]$, il *dominio di* A.
- Denotiamo i valori con $a, b, c, a_1, \ldots$

Relazioni e schemi

- L'*universo* U è un sottoinsieme finito di $\mathcal{U}$.
- Uno *schema di relazione* R è un sottoinsieme di U. Uno *schema di base di dati* D *su* U è un insieme di schemi di relazione con unione U:

$$U = \bigcup D = \bigcup \{R_1, R_2, \ldots, R_n\}.$$

- Sia D uno schema di base di dati su U, R uno schema di relazione e X un sottoinsieme di U. Per X-*tupla* intendiamo una funzione da X a $\cup_{A \in X} \Delta[A]$ tale che l'immagine di ogni A in X è un elemento di $\Delta[A]$. Una *relazione* r su R è un insieme finito di R-tuple. Una *base di dati* è un insieme di relazioni; una relazione su ogni schema di relazione di D.
- Usiamo: $D, D_1, \ldots$ per schemi di basi di dati; $R, R_1, \ldots$ per schemi di relazioni; $t, t_1, \ldots$ per tuple; $r, s, r_1, \ldots$ per relazioni; $d, d_1, \ldots$ per basi di dati; infine, $\alpha(r)$ per lo schema di relazione di r.

Algebra Relazionale

- Proposta da Codd nel 1970 come linguaggio di interrogazione per il modello relazionale (*Relational Algebra Query Language*)
- È composta da 6 operazioni: proiezione (π), join ($\bowtie$), unione ($\cup$), differenza ($-$), selezione (σ), sostituzione (ρ).
- La definizione di ciascuna operazione comprende tre parti:
  - restrizioni sugli schemi degli operandi
  - lo schema del risultato
  - la caratterizzazione del risultato

Proiezione di una tupla

- Sia t una X-tupla e $Z \subseteq X$. La *proiezione* di t su Z (denotata con $t[Z]$) è la Z-tupla t' tale che $t(A) = t'(A)$ per ogni $A \in Z$.
- **Esempio**: $X = \{A_1, A_2, A_3\}$, $t = \{(A_1, a), (A_2, 1), (A_3, b)\} \implies t[A_1] = \{(A_1, a)\}$, $t[A_1A_2] = \{(A_1, a), (A_2, 1)\}$
- Si noti che, formalmente, $t(A) \neq t[A]$. Ad esempio $t(A_1)$ è il *valore* a assunto dalla funzione t su A_1, invece $t[A_1]$ è la *tupla*, cioè una funzione, con dominio $\{A_1\}$, che associa ad A_1 il valore a.

Operazione di proiezione

- Operatore unario che, data una relazione r su R e un insieme di attributi $X \subseteq R$, produce una relazione $\pi_X(r)$ su X composta dalle proiezioni delle tuple di r su X.
  - $X \subseteq \alpha(r)$
  - $\alpha(\pi_X(r)) = X$
  - $\pi_X(r) = \{t[X] | t \in r\}$

Operazione di proiezione - Esempio

r

A_1	A_2	A_3
a	a	b
b	c	a
a	c	b
c	a	b
a	a	c

$\pi_{A_1 A_2}(r)$

A_1	A_2
a	a
b	c
a	c
c	a

$\pi_{A_2}(r)$

A_2
a
c

Nota. Intuitivamente, l'operazione di proiezione corrisponde a "cancellare" una o piú colonne dalla tabella che rappresenta l'operando, successivamente *eliminando eventuali righe duplicate*. (Infatti possono esistere due tuple distinte $t \in r$, $t' \in r$ tali che $t[X] = t'[X]$.

Operatore di selezione

- Operatore unario, con due parametri A, B, che devono essere attributi dello schema dell'operando, che produce una relazione composta da ogni tupla dell'operando che assume uguale valore nei parametri.
  - $A, B \in \alpha(r)$
  - $\alpha(\sigma_{A=B}(r)) = \alpha(r)$
  - $\sigma_{A=B}(r) = \{t : t \in r \text{ e } t(A) = t(B)\}$

r

A_1	A_2	A_3
b	b	a
c	b	c
a	c	b
a	c	a

$\sigma_{A_1=A_3}(r)$

A_1	A_2	A_3
c	b	c
a	c	a

Operatore di join

- Operatore binario. Il join $r \bowtie s$ è una relazione sull'unione degli schemi di r e s, contenente esattamente le tuple le cui proiezioni sugli schemi di r e s sono tuple di r e s, rispettivamente.
  - Nessuna restrizione agli schemi degli operandi
  - $\alpha(r \bowtie s) = \alpha(r) \cup \alpha(s)$
  - $r \bowtie s = \{t : t$ è una $(\alpha(r) \cup \alpha(s))$-tupla tale che $t[\alpha(r)] \in r$, $t[\alpha(s)] \in s\}$

r

A_1	A_2	A_3
c	a	a
a_1	c_1	b
a_1	b_1	b
b	a	a
c	b_1	a

s

A_2	A_3	A_4
a	a	b_1
b_1	b	c_1
b_1	b	a
b_1	a	c

$r \bowtie s$

A_1	A_2	A_3	A_4
c	a	a	b_1
a_1	b_1	b	c_1
a_1	b_1	b	a
b	a	a	b_1
c	b_1	a	c

La teoria delle dipendenze

- In generale non tutti i database con schema D corrispondono a un "mondo possibile" nel dominio modellato da D
- Ciò è dovuto alla presenza di *dipendenze* tra i dati.
  - I codici fiscali dei coniugi determinano univocamente la data del loro (primo) matrimonio; in qualunque relazione sullo schema $\{\texttt{CFMARITO}, \texttt{CFMOGLIE}, \texttt{DATA}\}$ non dovrebbero esistere due tuple con ugual proiezione su $\{\texttt{CFMARITO}, \texttt{CFMOGLIE}\}$ e differente proiezione su $\{\texttt{DATA}\}$.
  - La targa di un'auto determina ogni altra proprietà dell'auto; in qualunque relazione sullo schema $\{\texttt{TARGA}, \texttt{CILINDRATA}\}$ non dovrebbero esistere due tuple con ugual proiezione su $\{\texttt{TARGA}\}$ e differente proiezione su $\{\texttt{CILINDRATA}\}$
- Per questo motivo il modello relazionale è dotato, oltre che di operazioni, di *meccanismi di specifica* utilizzabili per definire l'insieme dei database possibili.
- La classificazione e lo studio dei meccanismi di specifica sono argomento della *teoria delle dipendenze* (*dependency theory*).

Dipendenze funzionali

- Si ricordi che con le ultime lettere maiuscole dell'alfabeto denotiamo insiemi di attributi.
- Una espressione della forma $X \longrightarrow Y$ è chiamata *dipendenza funzionale* e si legge "X determina funzionalmente Y", oppure "Y dipende funzionalmente da X".
- DEFINIZIONE Sia R uno schema di relazione, $X, Y \subseteq R$, e r una relazione su R. Diciamo che la dipendenza funzionale $X \longrightarrow Y$ *è soddisfatta da* r se e solo se

  non esistono tuple di r aventi la stessa proiezione su X e diversa proiezione su Y, ovvero, se due tuple qualunque di r hanno uguale proiezione su X, allora hanno uguale proiezione anche su Y. Formalmente:

$$\forall t_1, t_2 \in r : \; t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Dipendenze funzionali. Esempio

- Sia $R = \{\texttt{NOME}, \texttt{VIA}, \texttt{CITTA}, \texttt{CODICE}, \texttt{PREZZO}\}$, dove NOME è il nome di un fornitore, VIA e CITTA il suo indirizzo, CODICE il codice di un prodotto e PREZZO il prezzo attuale al quale il prodotto è fornito dal fornitore.
- Si assuma che, nel dominio rappresentato, non esistano fornitori con lo stesso nome e ogni fornitore abbia uno e un solo indirizzo.
- Conseguentemente è opportuno imporre, in ogni relazione r su R, la non esistenza di due tuple concordi su NOME ma discordi su VIA o CITTA.
- Inoltre, se si assume di memorizzare il prezzo attuale in PREZZO, e che non esistano prodotti diversi con lo stesso codice, allora ogni relazione r su R dovrebbe contenere al piú una sola tupla per ogni fissata coppia di valori di NOME e CODICE.
- Tali vincoli possono essere espressi mediante le seguenti dipendenze funzionali:

$$\texttt{NOME} \longrightarrow \{\texttt{VIA}, \texttt{CITTA}\}$$

$$\{\texttt{NOME}, \texttt{CODICE}\} \longrightarrow \texttt{PREZZO}.$$

Dipendenze funzionali. Esempio

NOME	VIA	CITTA	CODICE	PREZZO
Verdi	Garibaldi	Roma	01	1900
Bruni	Cavour	Roma	01	1930
Mori	Mazzini	Torino	08	4025
Galvani	Indipendenza	Bologna	03	7100
Verdi	Garibaldi	Roma	06	3400
Bruni	Cavour	Roma	03	7050
Verdi	Garibaldi	Roma	03	7300
Galvani	Indipendenza	Bologna	04	2750
Verdi	Garibaldi	Roma	08	3950
Verdi	Garibaldi	Roma	06	2360

- r soddisfa a $\texttt{NOME} \longrightarrow \{\texttt{VIA}, \texttt{CITTA}\}$.
- r non soddisfa a $\{\texttt{NOME}, \texttt{CODICE}\} \longrightarrow \texttt{PREZZO}$ (quinta e ultima tupla).

Istanze legali

- Sia $D = \{R_1, \ldots, R_n\}$ uno schema di base di dati su U e Σ un insieme di dipendenze su D, cioè dipendenze in cui ogni attributo è elemento di U.
- Uno schema e le dipendenze associate si indicano come la coppia (D, Σ).
- DEFINIZIONE Una base di dati d su D è una *istanza legale* di (D, Σ) se e solo se ogni relazione in d soddisfa a tutte le dipendenze in Σ.
- Chiamiamo *relazione legale* ogni relazione che è elemento di una istanza legale.
- **Osservazione**. Una dipendenza funzionale $X \longrightarrow Y$ può essere soddisfatta da ogni istanza legale di (D, Σ) senza che sia $X \longrightarrow Y \in \Sigma$; si dice che $X \longrightarrow Y$ è una *conseguenza* di Σ.

Forme normali

- Alcuni tra gli schemi di database su un assegnato universo U danno luogo ad *anomalie* legate all'aggiornamento dei dati e alla presenza di informazioni ridondanti
- La *teoria della normalizzazione* tratta i metodi per riconoscere e modificare gli schemi che presentano anomalie.
- Le anomalie si classificano in
  - anomalie di **modifica**
  - anomalie di **inserimento**
  - anomalie di **cancellazione**

Anomalie e ridondanza. Esempio

- Si assuma l'universo $U = \{\texttt{NOME}, \texttt{VIA}, \texttt{CITTA}, \texttt{CODICE}, \texttt{PREZZO}\}$ e $D = \{U\}$.

NOME	VIA	CITTA	CODICE	PREZZO
Verdi	Garibaldi	Roma	01	1900
Bruni	Cavour	Roma	01	1930
Mori	Mazzini	Torino	08	4025
Galvani	Indipendenza	Bologna	03	7100
Verdi	Garibaldi	Roma	06	3400
Bruni	Cavour	Roma	03	7050
Verdi	Garibaldi	Roma	03	7300
Galvani	Indipendenza	Bologna	04	2750
Verdi	Garibaldi	Roma	08	3950
Verdi	Garibaldi	Roma	04	2360

Anomalie e ridondanza. Esempio

1. L'indirizzo di un fornitore è replicato un numero (non noto a priori) di volte (ridondanza).
2. La modifica dell'indirizzo di un fornitore comporta la modifica dell'indirizzo in tutte le tuple in cui il fornitore compare (anomalia di modifica). Un errore in questa fase conduce alla incoerenza della base di dati.
3. Non si può inserire l'informazione relativa all'indirizzo di un fornitore se non vi è almeno un prodotto fornito dal fornitore (anomalia di inserimento).
4. Simmetricamente, se un fornitore cessa le forniture, non si può piú risalire al suo indirizzo (anomalia di cancellazione).
5. Lo schema della relazione può essere decomposto in due schemi $R_1 = \{\texttt{NOME}, \texttt{VIA}, \texttt{CITTA}\}$, $R_2 = \{\texttt{NOME}, \texttt{CODICE}, \texttt{PREZZO}\}$ con $D = \{R_1, R_2\}$.
6. Assumendo r su U, r_1 su R_1, r_2 su R_2, ogni base di dati $d = \{r\}$ si decompone in $d' = \{r_1, r_2\}$, e si ha $\pi_{R_1}(r) = r_1$, $\pi_{R_2}(r) = r_2$, $r = r_1 \bowtie r_2$.

Anomalie e ridondanza. Esempio

- Si assuma l'universo $U = \{\texttt{CODICE}, \texttt{NUMERO}, \texttt{NOME}, \texttt{CODICE\_MAN}\}$ e $D = \{U\}$.

CODICE	NUMERO	NOME	CODICE_MAN
01	2	Rossi	06
02	2	Bianchi	06
03	1	Neri	03
04	1	Grassi	03
05	3	Piccoli	05
06	2	Bassi	06
07	1	Storti	03
08	3	Fini	05

Anomalie e ridondanza. Esempio

- Il manager di un dipartimento è replicato un numero (non noto a priori) di volte (ridondanza).
- La modifica del manager di un dipartimento comporta la modifica del codice del manager in tutte le tuple in cui il dipartimento compare (anomalia di modifica).
- Un dipartimento e il suo manager non possono essere inseriti se non vi è almeno un dipendente nel dipartimento (anomalia di inserimento).
- Simmetricamente, se un dipartimento non ha piú dipendenti, il dipartimento e il suo manager vengono persi (anomalia di cancellazione).
- Lo schema della relazione può essere decomposto in due schemi $R_1 = \{\texttt{CODICE}, \texttt{NUMERO}, \texttt{NOME}\}$, $R_2 = \{\texttt{NUMERO}, \texttt{CODICE\_MAN}\}$ con $D = \{R_1, R_2\}$.
- Assumendo r su U, r_1 su R_1, r_2 su R_2, ogni base di dati $d = \{r\}$ si decompone in $d' = \{r_1, r_2\}$, e si ha $\pi_{R_1}(r) = r_1$, $\pi_{R_2}(r) = r_2$, $r = r_1 \bowtie r_2$.

Superchiavi e Chiavi

- Definizione (**Superchiave**)
  Dato uno schema R e un insieme di attributi $X \subseteq R$, X è una *superchiave* di R se e solo se, per ogni relazione legale r su R, si ha:

$$\forall t_1, t_2 \in r : \; t_1[X] = t_2[X] \implies t_1 = t_2.$$

- Si noti che, per qualunque R-tupla t, si ha $t = t[R]$. Quindi X è superchiave se e solo se, per ogni relazione legale r su R, R dipende funzionalmente da X.
- Definizione (**Chiave**)
  Dato uno schema R e un insieme di attributi $X \subseteq R$, X è una *chiave* di R se e solo se, per ogni relazione legale r dello schema si ha:
  1. X è superchiave
  2. non esiste Y tale che $Y \subset X$ e Y sia superchiave.

Superchiavi e chiavi: esempio

- **Esempio**: Dato lo schema e le dipendenze

$$R = \{\texttt{CODICE}, \texttt{NOME}, \texttt{NUMERO}, \texttt{CODICE\_MAN}\}$$

$$\Sigma = \{\texttt{CODICE} \longrightarrow \{\texttt{NOME}, \texttt{NUMERO}\},$$

$$\texttt{NUMERO} \longrightarrow \texttt{CODICE\_MAN}\}$$

- CODICE è superchiave
  - $t[\texttt{CODICE}] = t'[\texttt{CODICE}] \Rightarrow t[\texttt{NUMERO}] = t'[\texttt{NUMERO}]$ perché $\texttt{CODICE} \longrightarrow \{\texttt{NOME}, \texttt{NUMERO}\} \in \Sigma$
  - $t[\texttt{NUMERO}] = t'[\texttt{NUMERO}] \Rightarrow t[\texttt{CODICE\_MAN}] = t'[\texttt{CODICE\_MAN}]$ perché $\texttt{NUMERO} \longrightarrow \texttt{CODICE\_MAN} \in \Sigma$
  - $t[\texttt{CODICE}] = t'[\texttt{CODICE}] \Rightarrow t[\texttt{CODICE\_MAN}] = t'[\texttt{CODICE\_MAN}]$ per le precedenti
  - Dunque $t[\texttt{CODICE}] = t'[\texttt{CODICE}] \Rightarrow t = t'$
- CODICE è anche chiave (è un singolo attributo)
- $\{\texttt{CODICE}, \texttt{NOME}\}$ è superchiave (ma non chiave).

Attributi primi

- DEFINIZIONE (**Attributo primo**) Un attributo A dello schema R è *primo* se e solo se fa parte di almeno una chiave di R. In caso contrario A è detto non-primo.
- Nel seguito useremo per brevità la notazione $X \longrightarrow Y$ per indicare che la dipendenza $X \longrightarrow Y$ è vera in ogni istanza legale.
- Indicheremo una chiave mediante sottolineatura dei suoi attributi.

Prima e seconda forma normale

- La Prima Forma Normale (1NF) richiede solo che i domini degli attributi siano atomici.
- La seconda forma normale (2NF) è rivolta ad eliminare le anomalie che insorgono quando qualche attributo non primo dipende funzionalmente solo da una parte di una chiave (*dipendenza non completa* o *parziale*).
- **Esempio** Dato lo schema $\{\underline{\texttt{NOME}}, \texttt{VIA}, \texttt{CITTA}, \underline{\texttt{CODICE}}, \texttt{PREZZO}\}$ la dipendenza funzionale di CITTA da $\{\texttt{NOME}, \texttt{CODICE}\}$ non è completa, perchè si ha $\texttt{NOME} \longrightarrow \texttt{CITTA}$.
- DEFINIZIONE (**Seconda forma normale**) Uno schema R è in 2NF se ognuno dei suoi attributi non primi è completamente dipendente da ognuna delle chiavi.
- **Esempio** Lo schema $\{\underline{\texttt{NOME}}, \texttt{VIA}, \texttt{CITTA}, \underline{\texttt{CODICE}}, \texttt{PREZZO}\}$ non è in 2NF. Gli schemi $\{\underline{\texttt{NOME}}, \texttt{VIA}, \texttt{CITTA}\}$, $\{\underline{\texttt{NOME}}, \underline{\texttt{CODICE}}, \texttt{PREZZO}\}$ sono in 2NF.
- Se le chiavi sono tutte composte da un solo attributo, allora lo schema è sicuramente in 2NF.

Dipendenze transitive

- Nello schema $\{\underline{\texttt{CODICE}}, \texttt{NUMERO}, \texttt{NOME}, \texttt{CODICE\_MAN}\}$ con $\texttt{NUMERO} \longrightarrow \texttt{CODICE\_MAN} \in \Sigma$, valgono le dipendenze

$$\texttt{CODICE} \longrightarrow \texttt{NUMERO}$$

$$\texttt{CODICE} \longrightarrow \texttt{NOME}$$

$$\texttt{CODICE} \longrightarrow \texttt{CODICE\_MAN}$$

  Lo schema è in 2NF (la chiave è CODICE), ma vi sono ancora anomalie, dovute alle *dipendenze transitive*.

Dipendenze transitive

- DEFINIZIONE (**Dipendenza transitiva**) Dato lo schema S e $X \subseteq S$, $A \in S$, A è *transitivamente dipendente* da X se esiste $Y \subset S$ tale che:

$$X \longrightarrow Y,$$
$$Y \not\longrightarrow X,$$
$$Y \longrightarrow A$$
$$A \notin Y.$$

- **Esempio** Nello schema $\{\underline{\texttt{CODICE}}, \texttt{NUMERO}, \texttt{NOME}, \texttt{CODICE\_MAN}\}$ le condizioni della definizione sono verificate sostituendo `CODICE` a X, `NUMERO` a Y, `CODICE_MAN` a A. Dunque `CODICE_MAN` dipende transitivamente da `CODICE`.

Terza Forma Normale

- DEFINIZIONE (**Terza forma normale**) Uno schema R è in 3NF se e solo se ognuno dei suoi attributi non primi non dipende transitivamente da nessuna delle chiavi.
- **Esempio** Gli schemi:

$$\{\underline{\texttt{CODICE}}, \texttt{NUMERO}, \texttt{NOME}\}$$

$$\{\underline{\texttt{NUMERO}}, \texttt{CODICE\_MAN}\}$$

sono in 3NF.

Terza forma normale

- 2NF è una forma normale piú debole di 3NF:

  TEOREMA Se uno schema è in 3NF, allora è anche in 2NF.

  Dimostrazione. Supponiamo che R sia in 3NF ma non in 2NF. Allora esistono una chiave $X \subseteq R$, un suo sottoinsieme proprio Y e un attributo non primo $A \notin Y$, tali che $Y \longrightarrow A$. Poiché $Y \subset X$, si ha $X \longrightarrow Y$; inoltre se fosse $Y \rightarrow X$, si avrebbe $Y \rightarrow X \rightarrow R$, pertanto $Y \rightarrow R$, e X conterrebbe come sottoinsieme proprio una superchiave contro l'ipotesi che X sia chiave; quindi $Y \not\rightarrow X$. Dunque A è transitivamente dipendente da X, contro l'ipotesi che R sia in 3NF.