Definizioni

- Dati n domini $D_1, D_2 \dots D_n$ non necessariamente distinti, una **relazione** r è un sottoinsieme del prodotto cartesiano $D_1 \times D_2 \times \dots \times D_n$.
- Uno **schema di relazione** è un insieme di attributi.
- Uno **schema di base di dati** relazionale è un insieme $\{R_1, R_2 \dots R_n\}$ di schemi di relazione.
- Una *base di dati relazionale* con schema $\{R_1, R_2 \dots R_n\}$ è un insieme $\{r_1, r_2 \dots r_n\}$ dove r_i è un'**istanza** di relazione con schema R_i.
- Un'**anomalia** è un comportamento inaspettato e indesiderato della base di dati. Può essere di *inserimento*, *cancellazione*, *aggiornamento*, o *ridondanza*.
- Una **dipendenza funzionale** su uno schema di relazione R è una coppia ordinata di sottoinsiemi non vuoti $X, Y, \in R$ e viene denotata come $X \to Y$.
- Un'istanza r di R **soddisfa** la dipendenza funzionale $X \to Y$ se $\quad \forall(t1, t2) \in r, \quad t_1[X] = t_2[X] \quad \Rightarrow \quad t_1[Y] = t_2[Y]$
- Un'istanza r di R è **legale** se soddisfa tutte le dipendenze di un insieme F su R.
- La **chiusura** F^+ di un insieme di dipendenze funzionali F, è l'insieme di dipendenze funzionali che sono soddisfatte da ogni istanza legale di R.
- Un sottoinsieme K di R è una **chiave** per R se $K \to R \in F^+$, e se $\forall K' \subset K, K' \to R \notin F^+$.
- Un'attributo di R è **primo** se appartiene a una chiave di R.
- Un sottoinsieme X di R è una **superchiave** se contiene una chiave di R.
- Una dipendenza $X \to A \in F^+$ si dice **parziale** se A non è primo e X è contenuto propriamente in una chiave di R.
- Una dipendenza $X \to A \in F^+$ si dice **transitiva** se A non è primo e se $\forall$ chiave K si ha che X non è contenuto propriamente in K e $K - X \neq \emptyset$
- Uno schema di relazione R è in 3FN se, $\forall(X \to A) \in F^+ \mid A \notin X$, si ha che A è primo oppure X è una superchiave.
- **Assioma della riflessività**: $\quad Y \subseteq X \subseteq R \quad \Rightarrow \quad X \to Y \in F^A$
- **Assioma dell'aumento**: $\quad X \to Y \in F^A \quad \Rightarrow \quad XZ \to YZ \in F^A, \forall Z \subseteq R$
- **Assioma della transitività**: $\quad X \to Y \in F^A, Y \to Z \in F^A \quad \Rightarrow \quad X \to Z \in F^A$
- La **chiusura di** X rispetto a F, X^+_F, è definita nel modo seguente: $X^+_F = \{A \mid X \to A \in F^A\}$
- Una **decomposizione** di R è una famiglia $\rho = \{R_1, R_2 \dots R_k\}$ di sottoinsiemi di R che *ricopre* R $(\cup_{i=1}^k R_i = R)$
- Due insiemi di dipendenze funzionali F e G si dicono **equivalenti** se $F^+ = G^+$
- Una decomposizione ρ **preserva** F se $F \equiv \cup_{i=1}^k \pi_{R_i}(F)$, dove $\pi_{R_i}(F) = \{X \to Y \mid X \to Y \in F^+ \wedge XY \subseteq R_i\}$
- Una decomposizione ρ di R ha un **join senza perdita** se per ogni istanza legale di r di R si ha $r = \pi_{R_j}(F) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_k}(r)$
- Una **copertura minimale** di F è un insieme G di dipendenze equivalente a F tale che ogni dipendenza non è ridondante.

Teoremi

Teorema 1

Uno schema R è in 3FN se e solo se non esistono nè dipendene parziali nè dipendenze transitive in R.

Dim. *Parte solo se:* Banale. *Parte se:* Assurdo - R non è in 3FN; quindi esiste $X \to A$ tale che A non è primo e X non è superchiave. Ma $X \to A$ è perforza o parziale o transitiva.

Teorema 2

Regola dell'unione: $X \to Y \in F^A$ e $X \to Z \in F^A \quad \Rightarrow \quad X \to YZ \in F^A$

Dim. $XY \to YZ$ per aumento; $X \to XZ$ per aumento; $X \to YZ$ per transitività.

Regola della decomposizione: $X \to Y \in F^A$ e $Z \subseteq Y \quad \Rightarrow \quad X \to Z \in F^A$

Dim. $Y \to Z$ per riflessività; $X \to Z$ per transitività.

Regola della pseudotransitività: $X \to Y \in F^A$ e $WY \to Z \in F^A \quad \Rightarrow \quad WX \to Z \in F^A$

Dim. $WX \to WY$ per aumento; $WX \to Z$ per transitività.

Teorema 3

$F^+ = F^A$

Dim. *$F^+ \subseteq F^A$:* Sia $X \to Y \in F^A$. Dimostriamo $X \to Y \in F^+$ per induzione su Armstrong. Passo induttivo $(i = 0)$: $X \to Y \in F$, quindi $X \to Y \in F^+$. Induzione: per ipotesi, al passo $i - 1$, la dipendenza ottenuta è in F^+. Ora, con quale assioma abbiamo ottenuto $X \to Y$?: 1) Riflessività. $Y \subseteq X$; Se $t_1[X] = t_2[X]$, banalmente $t_1[Y] = t_2[Y]$. 2) Aumento. $X \to Y$ viene da $VZ \to WZ$. Presi t_1, t_2 tali che $t_1[X] = t_2[X]$, allora $t_1[V] = t_2[V]$ e $t_1[Z] = t_2[Z]$. Ma da V si ottiene $t_1[W] = t_2[W]$ per hp induttiva. E da W e da Z si ottiene $t_1[Y] = t_2[Y]$. 3) Transitiva. Ho che $X \to Z$ e $Z \to Y$. $t_1[X] = t_2[X]$ implica che $t_1[Z] = t_2[Z]$ che implica che $t_1[Y] = t_2[Y]$, per hp induttiva. *$F^+ \subseteq F^A$:* Assurdo. Diciamo che esiste una $X \to Y$ in F^+ ma non in F^A. Mostriamo che esiste un'istanza legale di R che non soddisfa $X \to Y$. Tabella con due righe: alla prima tutti 1; nella seconda tutti 1 fino alla fine di X^+, poi tutti 0. r è legale poichè se $V \to W$ non fosse legale, succederebbe che $V \subseteq X^+$ e $W \cap (R - X^+) \neq \emptyset$. Per Lemma 1 però $X \to V$, e per transitività $X \to W$, e di nuovo lemma 1 $W \subseteq X^+$, contraddizione. Infine r non soddisfa $X \to Y$. Se lo facesse, $Y \subseteq X^+$ per forza, poichè ci sono solo due possibili tuple e devono avere tutto coincidente. Ma per Lemma 1, $X \to Y \in F^+$, contraddizione.

Teorema 4

*L'**Algoritmo 1** calcola correttamente la chiusura di un insieme di attributi X rispetto a un insieme F di dipendenze.*

Dim. Si dimostrerà che, con j corrispondente a quando l'algoritmo termina, $A \in X^+ \Leftrightarrow A \in Z^j$ *Parte solo se:* Dimostriamo $\forall i \; Z^i \subseteq X^+$. Base induzione: $Z^0 = X$, $X \subseteq X^+$ per riflessione. Passo induttivo: $Z^{i-1} \subseteq X^+$ per hp induttiva; Prendiamo un attributo $A \in Z^i - Z^{i-1}$, che deve essere venuto da $Y \to V$ tale che $Y \subseteq Z^{i-1}$ e $A \in V$. $X \to Y$ per Lemma 1 poichè $Y \subseteq X^+$ per hp induttiva; $X \to V$ per transitività. Quindi $A \in X^+$. Quindi $Z^i \subseteq X^+$ *Parte se:* Prendiamo attributo $A \in X^+$. $X \to A$ deve essere soddisfatta in ogni istanza legale di R, per il Teorema 3. Prendiamo la stessa istanza r del Teorema 3, con gli 1 fino a Z^j e poi 0. r è legale poichè se ci fosse $V \to W$ non soddisfatta, allora $V \subseteq Z^j$ e $W \cap (R - Z^j) \neq \emptyset$, ma in tal caso si avrebbe $S^j \not\subseteq Z^j$, contraddizione. Ora, siccome deve essere soddisfatta anche $X \to A$, significa che $A \in Z^j$, poichè $X = Z^0 \subseteq Z^j$, cioè deve stare tra gli uni.

Teorema 5

*L'**Algoritmo 3** calcola correttamente X^+_G, dove $G = \cup_{j=1}^k \pi_{R_j}(F)$*

Dim. Si dimostrerà che, con f corrispondente a quando l'algoritmo termina, $A \in Z^f \Leftrightarrow A \in X^+_G$. *Parte solo se*: Base induzione: $Z^0 = X, X \subseteq X^+$, quindi $Z^0 \subseteq X^+_G$. Passo induttivo: Sia $A \in Z^i - Z^{i-1}$, allora $\exists$ un indice $j \mid A \in (Z^{i-1} \cap R_j)^+_F \cap R_j$. Si ha che $(Z^{i-1} \cap R_j) \to A \in F^+$ (per teorema 3). Per la definizione di G, e poichè $A \in R_j$ e $Z^{i-1} \cap R_j \subseteq R_j$, $(Z^{i-1} \cap R_j) \to A \in G$. Per hp induttiva, $X \to (Z^{i-1} \in G^+$, per decomposizione, $X \to (Z^{i-1} \cap R_j) \in G^+$, per transitività $X \to A \in G^+$, cioè $A \in X^+_G$, quindi $Z^i \subseteq X^+_G$. *Parte se:* Useremo la proposizione che $X \subseteq Y \to X^+_F \subseteq Y^+_F$. Siccome $X = Z^0 \subseteq Z^f$, dalla proposizione $X^+_G \subseteq (Z^f)^+_G$. Dimostreremo che $Z^f = (Z^f)^+_G$. Eseguiamo l'algoritmo 1, dando come input Z^f e G. Se per assurdo $Z^f \neq (Z^f)^+_G$, deve esistere $B \in S^0 \not\in Z'^0$. Siccome si ha che $S^0 = \{A \mid (Y \to V \in G) \wedge (A \in V) \wedge (Y \subseteq Z'^0)\}$, per la definizione di G, deve esistere $j \mid B \in \{A \mid (Y \to V \in F^+) \wedge (A \in V) \wedge (Y \subseteq Z'^0) \wedge (YV \subseteq R_j)\}$. Da $Y \to V \in F^+$, per il Lemma 1, $V \subseteq Y^+_F$. Inoltre, siccome $Y \subseteq Z'^0 \wedge YV \subseteq R_j$, $Y \subseteq Z'^0 \cap R_j = Z^f \cap R_j$, si ha che $V \subseteq (Z^f \cap R_j)^+_F$, proprio per la proposizione. Infine, poichè $YV \subseteq R_j$, $V \subseteq (Z^f \cap R_j)^+_F \cap R_j$. Ma $B \in V$ e quindi $B \in (Z^f \cap R_j)^+_F \cap R_j$ e quindi $B \in S^f$. E siccome $B \notin Z^f$, questo significa che Z^f non è il valore finale di Z (contraddizione).

Teorema 6

$m_p(r) = \pi_{R_1}(r) \bowtie \dots \bowtie \pi_{R_k}(r)$. *Si ha che:*

- $r \subseteq m_p(r)$

Dim. Sia t una tupla di r. $\forall i \in \{1 \dots k\}, t[R_i] \in \pi_{R_i}(r)$, e quindi $t \in m_\rho(r)$

- $\pi_{R_i}(m_p(r)) = \pi_{R_i}(r)$

Dim. Per la precedente $r \subseteq m_\rho(r)$, quindi $\pi_{R_i}(r) \subseteq \pi_{R_i}(m_\rho(r))$. Mostriamo che $\pi_{R_i}(r) \supseteq \pi_{R_i}(m_\rho(r))$ poichè $\forall$ tupla $t \in m_\rho(r)$ e $\forall i \in \{1 \dots k\}$, $\exists t' \mid t[R_i] = t'[R_i]$.

- $m_p(m_p(r)) = m_p(r)$

Dim. Per la precedente $\pi_{R_i}(m_\rho(r)) = \pi_{R_i}(r)$. Allora $m_\rho(m_\rho(r)) = \pi_{R_1}(m_\rho(r)) \bowtie \dots \bowtie \pi_{R_k}(m_\omega(r)) = \pi_{R_1}(r) \bowtie \dots \bowtie \pi_{R_k}(r) = m_\rho(r)$.

Teorema 7

*L'**Algoritmo 4** decide correttamente se ρ ha un join senza perdita.*

Dim. Bisogna dimostrare che ρ ha un join senza perdita $\Leftrightarrow$ la tabella r ha una riga con tutte a. *Parte solo se*: La tabella r può essere interpretata come un'istanza legale di R, poichè l'algoritmo termina quando non ci sono più violazioni delle dipendenze in F. Inoltre, non viene modificata nessuna a. Se per assurdo ρ ha un join senza perdita ma la tabella non ha una riga con tutte a, allora $\forall i \in \{1 \dots k\}$, $\pi_{R_i}(r)$ contiene una riga con tutte a; allora $m_\rho(r)$ contiene una riga con tutte a, e quindi $m_\rho(r) \neq r$ (contraddizione).

Teorema 8

*L'**Algoritmo 5** permette di calcolare in tempo polinomiale una decomposione ρ tale che ogni schema in ρ è in 3FN e che ρ preserva F*

Dim. *ρ preserva F*: Sia $G = \cup_{i=1}^k \phi_{R_i}(F)$. Siccome $\forall X \to A \in F$ vale che $XA \in \rho$, si ha che $G \supseteq F$, e $G^+ \supseteq F^+$. Inoltre, $G^+ \supseteq F^+$ siccome per definizione $G \supseteq F^+$. *Ogni schema in ρ è in 3FN*: Prendiamo $S \in \rho$. Ogni attributo in S fa parte della chiave, e quindi S è in 3FN (Ricorda che S è l'insieme degli attributi non contenuti in nessuna dipendenza funzionale). Se $R \in \rho$ vuol dire che c'è una dipendenza funzionale che coinvolge tutti gli attributi di R. Sarà della forma $R - A \to A$ poichè F è una copertura minimale; e $R - A$ è chiave in R. Sia $Y \to B \in F^+$; se $B = A$ allora $Y = R - A$, poichè F è copertura minimale, e quindi Y è superchiave; se $B \neq A$ allora $B \in R - A$ e quindi B è primo. Se $XA \in \rho$, non esiste $X' \to A \in F^+ \mid X' \supset X$ poichè F è copertura minimale, e quindi X è chiave in XA. Sia $Y \to B \in F^+ \mid YB \supseteq XA$; se $B = A$, allora $Y = X$, cioè Y è superchiave, poichè F è copertura minimale; se $B \neq A$, allora $B \in X$ e quindi B è primo.

Teorema 9

La decomposione $\sigma = \rho \cup \{K\}$, dove K è una chiave, è tale che ogni schema in σ è in 3FN, σ preserva F, e che σ ha un join senza perdita.

Dim. *σ preserva F*: Sia $\sigma = \{R_1 \dots R_{k+1}\}$ dove $\rho = \{R_1 \dots R_k\}$ è ottenuta con Algoritmo 5 e $R_{k+1} = K$. Sia $G' = \cup_{i=1}^{k+1}\{X \to Y \in F^+ \mid XY \in R_i\}$, e $G = \cup_{i=1}^{k}\{X \to Y \in F^+ \mid XY \in R_i\}$. Per Teorema 8 $F \equiv G$; $G \subseteq G'$ e quindi $G \subseteq G'^+$. Per definizione, $G' \subseteq F^+$, e $G' \subseteq G^+$ poichè $F^+ = G^+$. Per il lemma 2, siccome $G \subseteq G'$ e $G' \subseteq G^+$, allora $G \equiv G'$. *Ogni schema in ρ è in 3FN*: è sufficiente mostrare che K è in 3FN, poichè $\sigma = \rho \cup \{K\}$. Se K non fosse chiave di K, esisterebbe $K' \subset K \mid K' \to K \in F^+$. Ma $K \to R$ poichè K è chiave di R per definizione, e per transitività $K' \to R \in F^+$, che contraddice il fatto che K è chiave di R. *σ ha un join senza perdita*: Mostriamo che viene prodotta una tabella con tutte a dall'algoritmo 4. Supponiamo che l'algoritmo esamini le dipendenze $Y_1 \to A_1 \dots Y_n \to A_n$ dove A_i è l'i-esimo attributo esaminato dall'Algoritmo 1 per calcolare K^+, mentre $Y_i \subseteq Z^{i-1} = KA_1, \dots A_{i-1} \subseteq K^+$. Base induzione: Poichè $Y_1 \subseteq Z^0 = K$, sia nella riga dello schema Y_1A_1 che in quella di K ci sono tutte a in corrispondenza degli attributi di Y_1, e c'è una a in corrispondenza di A_1 nella riga Y_1A_1. Allora l'algoritmo 4 mette una a nella riga di K in corrispondenza di A_1. Induzione: Per hp induttiva, nella riga di K c'è una a $\forall j \mid j \leq i - 1$. Poichè $Y_i \subseteq KA_1 \dots A_{i-1}$, nella riga di Y_iA_i e nella riga di K ci sono tutte a in corrispondenza degli attributi in Y_i; inoltre nella riga di Y_iA_i c'è una a in corrispondenza di A_i. Allora l'algoritmo 4 mette una a in corrispondenza di A_i nella riga di K. Quindi alla fine nella riga di K ci saranno tutte a.

Lemma 1

- $Y \subseteq X^+ \Leftrightarrow X \to Y \in F^A$

  Dim. Sia $Y = A_1 \dots A_n$; *Parte se:* $\forall i X \to A_i$ poichè $Y \subseteq X^+$; per unione, $X \to Y$. *Parte solo se:* $X \to Y$; per decomposizione $\forall i X \to A_i$. Quindi $A_1 \dots A_n \subseteq X^+$ e $Y \subseteq X^+$

Lemma 2

- $F \subseteq G^+ \;\Rightarrow\; F^+ \subseteq G^+$

Algoritmi

Algoritmo 1

$Z := X$
$S := \{A \mid (Y \to V \in F) \wedge (A \in V) \wedge (Y \subseteq Z)\}$;
while $S \not\subseteq Z$
    $Z := Z \cup S$;
    $S := \{A \mid (Y \to V \in F) \wedge (A \in V) \wedge (Y \subseteq Z)\}$;

Algoritmo 2

$successo = true$;
for each $X \to Y \in F$
    calcola X^+_G;
    if $Y \not\subseteq X^+_G$ then $successo = false$;

Algoritmo 3

$Z := X$;
$S := \emptyset$;
for $j := 1$ to k
    $S := S \cup [(Z \cap R_j)^+_F \cap R_j]$;
while $S \not\subseteq Z$
    $Z := Z \cup S$;
    for $j := 1$ to k
        $S := S \cup [(Z \cap R_j)^+_F \cap R_j]$;

Algoritmo 4

Costruisci una tabella con $|R|$ colonne e $|\rho|$ righe. All'incrocio dell'i-esima riga e della j-esima colonna, si metta a_j se $A_j \in R_i$, altrimenti $b_{i,j}$
repeat
for each $X \to Y \in F$
    if $\exists\{t1, t2\} \in r \mid t_1[X] = t_2[X] \wedge t_1[Y] \neq t_2[Y]$ then
        for each $A_j \in Y$
            if $t_1[A_j] = "a_j"$ then $t_2[A_j] := t_1[A_j]$;
            else $t_1[A_j] := t_2[A_j]$;
until r ha una riga con tutte "a" oppure r non è cambiato
if r ha una riga con tutte "a", allora ρ ha un join senza perdita.

Algoritmo 5

$S := \emptyset$;
for each $A \in R$ tale che A non è in nessuna dipendenza in F
    $S := S \cup \{A\}$;
if $S \neq \emptyset$ then
    $R := R - S$;
    $\rho := \rho \cup \{S\}$;
if esiste una dipendenza in F che coinvolge tutti gli attributi in R then
else for each $X \to A \in F$ do $\rho := \rho \cup \{XA\}$;
    $\rho := \rho \cup \{R\}$;

Anomalie

Anomalie: anomalie di inserimento, di cancellazione, di aggiornamento. Ridondanza dei dati.

Da rivedere

Da rivedere la definizinoe della copertura minimale.

Organizzazione dei File

File Heap

-Inserimento: 1 accesso in lettura (leggo l'ultimo blocco); 1 accesso scrittura (scrivi il blocco modificato, chiedine uno nuovo al fyle system se pieno)
-Ricerca: sequenziale blocco per blocco. Da 1 a $numblocchi$ accessi. Costo medio: $(R + 2R \dots + nR)/N = R/N * n(n+1)/2 = n/2$ con N numero totale di record, R record per blocco.
-Modifica: Ricerca + 1 accesso scrittura.
-Cancellazione: Ricerca + 1 accesso lettura + 2 accessi scrittura (ultimo + modificato)

File Hash

-Ricerca: $1/B$ esimo della ricerca su un heap (assumendo una buona funzione hash, come esempio divido i bit della chiave in gruppi lunghi uguali, li sommi in binario e fai il modulo per B)

File isam (Indexed Sequential Access Method)

-Ci sono file indice e file principali
-Ricerca: si cerca nel FI un record con valore K' che ricopre K, cioè che $K' \leq K$. Costo: $\lceil log_2(numblocchi) \rceil + 1$.
-Ricerca per interpolazione: serve una funzione f che presi K_1, K_2, K_3, restituisce la frazione dell'intervallo fra K_2 e K_3 dove sta K_1. Costo: $\lceil log_2 log_2(numblocchi) \rceil + 1$.
-Inserimento: costo ricerca + 1 se sul blocco c'è spazio, altrimenti di più. Se sono pieni sia quello prima che quello dopo, bisogna chiederne uno nuovo e ripartire i record tra vecchio e nuovo.
-Cancellazione: costo ricerca + 1. Se il record è il primo del blocco, altri accessi necessari (restituzione blocco a file system e modifica FI)
-Modifica: costo ricerca + 1 se la chiave non cambia, altrimenti costo cancellazione + inserimento.

File B - Tree

-Ricerca: $h + 1$, con h altezza dell'albero. Massimo valore di $h = log_d(N/e)$, con $2e - 1$ record per blocco FP, $2d - 1$ record per blocco FI, e N numero record di FP.
-Inserimento: costo ricerca + 1 se c'è spazio, altrimenti costo ricerca + 1 + s, con $s \leq (2h + 1)$ (scrittura blocchi FI in risalita)
-Cancellazione: costo ricerca + 1 se il blocco rimane pieno almeno a metà. Altrimenti più accessi.
-Modifica: costo ricerca + 1 se non cambia la chiave, altrimenti cancellazione + inserimento.