

# Basi di Dati

## Soluzioni Esercizi: Calcolo Relazionale e Datalog

Marco Abbadini  
marco.abbadini@unibg.it  
<https://cs.unibg.it/abbadini>

5 aprile 2023

### 1 Calcolo Relazionale

UTENTE(Codice, Nome, Cognome, Indirizzo, Telefono)  
PRESTITO(Collocazione, CodUtente, DataPrestito, DataRest)  
COPIA(Collocazione, ISBN, DataAcq)  
DATILIBRO(ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

1. Trovare i titoli di tutti i libri pubblicati negli anni '80
2. Trovare i titoli di tutti i libri non pubblicati negli anni '80
3. Trovare i titoli dei libri di informatica prestati nel giugno '02
4. Estrarre nome, cognome e codice degli utenti che non hanno mai preso in prestito libri di informatica

#### Es. 1

Trovare i titoli di tutti i libri pubblicati negli anni '80

$$\{t \mid \exists t_1 \in DATILIBRO (t[Titolo] = t_1[Titolo] \wedge t_1[AnnoPub] \geq 1980 \wedge t_1[AnnoPub] < 1990)\}$$

#### Es. 2

Trovare i titoli di tutti i libri non pubblicati negli anni '80

$$\{t \mid \exists t_1 \in DATILIBRO (t[Titolo] = t_1[Titolo] \wedge t_1[AnnoPub] < 1980 \wedge t_1[AnnoPub] \geq 1990)\}$$

Oppure

$$\{t \mid \exists t_1 \in DATILIBRO (t[Titolo] = t_1[Titolo] \wedge \neg (t_1[AnnoPub] \geq 1980 \wedge t_1[AnnoPub] < 1990))\}$$

#### Es. 3

Trovare i titoli dei libri di informatica prestati nel giugno '02

$$\{t \mid \exists t_1 \in DATILIBRO, \exists t_2 \in COPIA, \exists t_3 \in PRESTITO (t[Titolo] = t_1[Titolo] \wedge t_1[Genere] = Informatica \wedge t_1[ISBN] = t_2[ISBN] \wedge t_2[Collocazione] = t_3[Collocazione] \wedge t_3[DataPr] \geq '01062022' \wedge t_3[DataPr] < '01072002')\}$$

#### Es. 4

Estrarre nome, cognome e codice degli utenti che non hanno mai preso in prestito libri di informatica

$$\{t \mid \exists t_1 \in UTENTE(t[Nome] = t_1[Nome] \wedge t[Cognome] = t_1[Cognome]) \wedge \\ t[Codice] = t_1[Codice] \wedge \neg(\exists t_2 \in DATILIBRO, \exists t_3 \in COPIA, \exists t_4 \in PRESTITO \\ (t_4[CodUtente] = t_1[Codice] \wedge t_4[Collocazione] = t_3[Collocazione] \wedge \\ t_3[ISBN] = t_2[ISBN] \wedge t_2[Genere] = 'Informatica'))\}$$

## 2 Datalog I

GENITORE(Padre, Figlio)

1. Ricavare tutti i discendenti di ogni padre
2. Ricavare tutti i discendenti di Bob
3. Ricavare tutti i discendenti di Bob ma non di Alice

#### Es. 1

Ricavare tutti i discendenti di ogni padre

$$R(x, y) : - GENITORE(x, y) \\ R(x, z) : - R(x, y), GENITORE(y, z)$$

#### Es. 2

Ricavare tutti i discendenti di Bob

$$DISCENDENTEBOB(y) : - GENITORE('Bob', y) \\ DISCENDENTEBOB(x) : - DISCENDENTEBOB(y), GENITORE(y, x)$$

#### Es. 3

Ricavare tutti i discendenti di Bob ma non di Alice

$$DISCALICE(x) : - GENITORE('Alice', x) \\ DISCALICE(x) : - DISCALICE(y), GENITORE(y, x) \\ DISCBOBNOTALICE(x) : - GENITORE('Bob', x), NOT DISCALICE(x) \\ DISCBOBNOTALICE(x) : -DISCBOBNOTALICE(y), GENITORE(y, x), NOT DISCALICE(x)$$

La seguente soluzione è invece sbagliata:

$$R(x, y) : - GENITORE('Bob', y) \\ R(x, z) : - R(x, y), NOTGENITORE('Alice', z)$$

in quanto la sconda regola non è safe, z compare solo una volta ed è negato. (Si vedano le slide di teoria)

## 3 Datalog II

AMICO(Persona, Persona)

NEMICO(Persona, Persona)

1. Trovare tutti gli amici e gli amici degli amici di Bob (attenzione alla terminazione)
2. Trovare tutti gli amici di Bob che non hanno amici eccetto Bob
3. Trovare tutte quelle persone che hanno come amici i nemici dei loro amici

### Es. 1

Trovare tutti gli amici e gli amici degli amici di Bob (attenzione alla terminazione)

$$R(x) : - AMICO('Bob', x)$$
$$R(x) : - AMICO('Bob', z), AMICO(z, x)$$

### Es. 2

Trovare tutti gli amici di Bob che non hanno amici eccetto Bob

$$AMICIBOB(x) : - AMICO('Bob', x)$$
$$ESCLUSI(x) : - AMICOB(x), AMICO(x, y), y \neq 'Bob'$$
$$R(x) : - AMICOB(x), NOT ESCLUSI(x)$$

### Es. 3

Trovare tutte quelle persone che hanno come amici i nemici dei loro amici  
Inserire nel risultato anche tutti coloro che non hanno nemici né amici

Tutte le persone che non hanno amici:

$$AMICIASX(x) : - AMICO(x, y)$$
$$TUTTI1(x) : - AMICO(x, y)$$
$$TUTTI1(x) : - AMICO(y, x)$$
$$AMICISOLOaDX(x) : - TUTTI1(x), NOT AMICIASX(x)$$

Tutte le persone che non hanno nemici:

$$NEMICIASX(x) : - NEMICO(x, y)$$
$$TUTTI2(x) : - NEMICO(x, y)$$
$$TUTTI2(x) : - NEMICO(y, x)$$
$$NEMICISOLOaDX(x) : - TUTTI2(x), NOT NEMICIASX(x)$$

Risultato:

$$R(x) : - NEMICO(x, y), NEMICO(y, z), AMICO(x, z)$$
$$R(x) : - NEMICISOLOaDX(x)$$
$$R(x) : - AMICISOLOaDX(x)$$