

# Algebra relazionale

## Esercizi con soluzione

---

Lucia Ferrari

lucia02.ferrari@edu.unife.it

## Esercizi con soluzione

---

Sia dato il seguente schema relazionale:

*MOTO*(targa, cilindrata, marca, nazione, tasse)

*PROPRIETARIO*(nome, targa).

Esprimere in algebra relazionale le seguenti interrogazioni (usando le funzioni aggregate solo se necessario e senza usare l'operatore di divisione  $\div$ ).

---

<sup>1</sup>Crediti: Prof. Angelo Montanari, corso di Basi di Dati per la Laurea Triennale, Università di Udine.

## Esercizio 1

1. Determinare le persone che possiedono solo moto della stessa marca.

$$\begin{aligned}MOTO\_POSS1 &\leftarrow \pi_{nome, marca}(MOTO * PROPRIETARIO) \\MOTO\_POSS2(nome1, marca1) &\leftarrow MOTO\_POSS \\NO\_GOOD &\leftarrow \pi_{nome}(MOTO\_POSS1 \\&\quad \bowtie_{nome=nome1 \wedge marca \neq marca1} MOTO\_POSS2) \\RISULTATO &\leftarrow \pi_{nome}(PROPRIETARIO) - NO\_GOOD\end{aligned}$$

2. Determinare i nomi delle persone che possiedono solo moto giapponesi di almeno due marche diverse.

$$\begin{aligned}NO\_GOOD &\leftarrow \pi_{nome}(\sigma_{nazione \neq "Giappone"}(MOTO * PROPRIETARIO)) \\SOLO\_GIAPPO &\leftarrow \pi_{nome}(PROPRIETARIO) - NO\_GOOD \\ALMENO2\_M &\leftarrow \pi_{nome}(MOTO\_POSS1 \bowtie_{nome=nome1 \wedge marca \neq marca1} MOTO\_POSS2) \\RISULTATO &\leftarrow SOLO\_GIAPPO \cap ALMENO2\_M\end{aligned}$$

Le relazioni  $MOTO\_POSS1$  e  $MOTO\_POSS2$  sono quelle del punto 1.

2. Determinare i nomi delle persone che possiedono solo moto giapponesi di almeno due marche diverse.

targa	cilindrata	marca	nazione	tasse
ABC123	1000	Yamaha	Giapponese	100
DEF456	750	Suzuki	Giapponese	80
GHI789	1200	Honda	Giapponese	120
JKL012	850	Ducati	Italiana	90
MNO345	1100	BMW	Tedesca	110
PQR678	650	Kawasaki	Giapponese	70
STU901	800	Aprilia	Italiana	85
VWX234	900	Yamaha	Giapponese	95

Table 1: Tabella Moto

nome	targa
Mario Rossi	ABC123
Mario Rossi	DEF456
Paolo Verdi	GHI789
Roberto Neri	JKL012
Giuseppe Rossi	MNO345
Paolo Verdi	PQR678
Anna Grigi	VWX234
Giuseppe Rossi	STU901

Table 2: Tabella Proprietario

- No\_Good: trovo i proprietari che hanno moto non giapponesi

$$NO\_GOOD \leftarrow \pi_{nome}(\sigma_{nazione \neq "Giappone"}(MOTO * PROPRIETARIO))$$

nome
Roberto Neri
Giuseppe Rossi

Table 3: Tabella NO\_GOOD

- SOLO\_GIAPPO: trovo i proprietari che hanno solo moto giapponesi

$$NO\_GOOD \leftarrow \pi_{nome}(\sigma_{nazione \neq "Giappone"}(MOTO * PROPRIETARIO))$$

nome
Mario Rossi
Paolo Verdi
Anna Grigi

Table 4: Tabella SOLO\_GIAPPO

- MOTO\_PROPR1 e MOTO\_PROPR2: trovo le le marche associate ad ogni proprietario

nome	marca
Mario Rossi	Yamaha
Mario Rossi	Suzuki
Paolo Verdi	Honda
Roberto Neri	Ducati
Giuseppe Rossi	BMW
Paolo Verdi	Kawasaki
Anna Grigi	Yamaha
Giuseppe Rossi	Aprilia

Table 5: MOTO\_PROPR1

nome1	marca1
Mario Rossi	Yamaha
Mario Rossi	Suzuki
Paolo Verdi	Honda
Roberto Neri	Ducati
Giuseppe Rossi	BMW
Paolo Verdi	Kawasaki
Anna Grigi	Yamaha
Giuseppe Rossi	Aprilia

Table 6: MOTO\_PROPR2

- ALMENO2\_M: trovo i proprietari che possiedono moto con almeno due marche

$$ALMENO2\_M \leftarrow \pi_{nome}(MOTO\_POSS1 \bowtie_{nome=nome1 \wedge marca \neq marca1} MOTO\_POSS2)$$

nome
Mario Rossi
Paolo Verdi
Giuseppe Rossi

Table 7: Tabella ALMENO2\_M

Risultato: faccio l'intersezione tra la tabella solo giapponesi e almeno 2 marche per trovare i nomi dei proprietari che soddisfano entrambe le condizioni

nome
Mario Rossi
Paolo Verdi
Anna Grigi

Table 8: Tabella SOLO\_GIAPPO

nome
Mario Rossi
Paolo Verdi
Giuseppe Rossi

Table 9: Tabella ALMENO2\_M

nome
Mario Rossi
Paolo Verdi

Table 10: Risultato



## Esercizio 1

3. Determinare le tasse che ogni proprietario deve pagare per le moto possedute (si assuma che una persona possa possedere più moto, ma che ogni moto sia posseduta da una sola persona).

$$R \leftarrow \text{nome} \mathcal{F}_{SUM(tasse)}(PROPRIETARIO * MOTO)$$

4. Individuare la nazione (le nazioni, se più d'una soddisfa le condizioni) in cui è prodotto il maggior numero di moto fra quelle registrate nella base di dati (si assuma che l'attributo *nazione* di *MOTO* designi la nazione che produce la moto).

$$\begin{aligned} MOTONAZ1(NAZ1, NUM1) &\leftarrow \text{nazione} \mathcal{F}_{COUNT(targa)}(MOTO) \\ MOTONAZ2(NAZ2, NUM2) &\leftarrow MOTONAZ1 \\ NO\_GOOD &\leftarrow \pi_{NAZ1}(MOTONAZ1 \bowtie_{NAZ1 \neq NAZ2 \wedge NUM1 < NUM2} MOTONAZ2) \\ R &\leftarrow \pi_{\text{nazione}}(MOTO) - NO\_GOOD \end{aligned}$$

5. Determinare le persone che possiedono moto della cilindrata minima (fra le cilindrature presenti nella base di dati).

$$\begin{aligned} \text{PROPR1}(\text{NOME1}, \text{CIL1}) &\leftarrow \pi_{\text{n\acute{o}me}, \text{cilindrata}}(\text{PROPRIETARIO} * \text{MOTO}) \\ \text{PROPR2}(\text{NOME2}, \text{CIL2}) &\leftarrow \text{PROPR1} \\ \text{NO\_GOOD} &\leftarrow \pi_{\text{NOME1}}(\text{PROPR1} \bowtie_{\text{NOME1} \neq \text{NOME2} \wedge \text{CIL1} > \text{CCIL2}} \text{PROPR2}) \\ R &\leftarrow \pi_{\text{n\acute{o}me}}(\text{PROPRIETARIO}) - \text{NO\_GOOD} \end{aligned}$$

NO\_GOOD = Prendo tutti i proprietari che hanno cilindrata NON minima

Sia dato il seguente schema relazionale:

*FORNITORE*(*S#*, *fnome*, *status*, *citta'*)

*COMPONENTE*(*C#*, *cnome*, *colore*, *peso*, *citta'*)

*PROGETTO*(*P#*, *pnome*, *citta'*)

*FORNISCE*(*S#*, *C#*, *P#*, *quantita'*).

Definiamo *originale* un componente fornito da un solo fornitore.

Esprimere in algebra relazionale le seguenti interrogazioni, ricorrendo alle funzioni aggregate solo se necessario e senza usare l'operatore di divisione  $\div$ .

---

<sup>2</sup>Crediti: Prof. Angelo Montanari, corso di Basi di Dati per la Laurea Triennale, Università di Udine.

## Un esempio di possibili tabelle

S#	fnome	status	città
S1	Fornitore1	102	Roma
S2	Fornitore2	104	Milano
S3	Fornitore3	92	Napoli
S4	Fornitore4	80	Firenze

Table 11: Tabella FORNITORE

C#	cnome	colore	peso	città
C1	Componente1	Rosso	10	Roma
C2	Componente2	Blu	20	Milano
C3	Componente3	Verde	30	Napoli
C4	Componente4	Giallo	40	Firenze
C5	Componente5	Giallo	10	Firenze
C6	Componente6	Nero	4	Roma

Table 12: Tabella COMPONENTE

P#	pnome	città
P1	Progetto1	Roma
P2	Progetto2	Milano
P3	Progetto3	Napoli
P4	Progetto4	Firenze

Table 13: Tabella PROGETTO

S#	C#	P#	quantità
S1	C1	P1	100
S2	C2	P1	200
S3	C2	P2	150
S3	C3	P3	50
S4	C4	P4	20
S3	C5	P1	50
S4	C6	P4	210
S3	C6	P1	10

Table 14: Tabella FORNISCE

## Esercizio 2

1. Determinare i nomi dei fornitori che non forniscono alcun componente originale.

$$\begin{aligned} \text{FORN} &\leftarrow \pi_{S\#, C\#}(\text{FORNISCE}) \\ \text{FORN1}(S\#1, C\#1) &\leftarrow \text{FORN} \\ \text{FORN\_COMP\_NON\_OR} &\leftarrow \pi_{S\#, C\#}(\text{FORN} \bowtie_{S\# \neq S\#1 \wedge C\# = C\#1} \text{FORN1}) \\ \text{NO\_GOOD} &\leftarrow \pi_{S\#}(\text{FORN} - \text{FORN\_COMP\_NON\_OR}) \\ \text{GOOD} &\leftarrow \pi_{S\#}(\text{FORN}) - \text{NO\_GOOD} \\ R &\leftarrow \pi_{nome}(\text{FORNITORE} * \text{GOOD}) \end{aligned}$$

FORN\_COMP\_NON\_OR: trovo i fornitori che offrono dei componenti non originali (non so se siano tutti non originali o solo alcuni)

NO\_GOOD: dalle tuple (codice fornitore, codice componente) dei fornitori rimuovo le tuple (codice fornitore, codice componente non originale) ottenendo come risultato (codice fornitore, codice componente originale), ovvero i nomi dei produttori che offrono DELLE componenti originali -> quelle che non vogliamo

GOOD: trovo i fornitori che non offrono componenti originali

R: trovo il nome dei fornitori trovati precedentemente

2. Determinare i nomi dei fornitori che forniscono solo componenti originali.

$$\begin{aligned} NO\_GOOD &\leftarrow \pi_{S\#}(FORN \bowtie_{S\# \neq S\#1 \wedge C\# = C\#1} FORN1) \\ GOOD &\leftarrow \pi_{S\#}(FORN) - NO\_GOOD \\ R &\leftarrow \pi_{fname}(FORNITORE * GOOD) \end{aligned}$$

NO\_GOOD = otteniamo i nomi dei produttori che forniscono delle componenti NON originali (relazioni FORN e FORN1 sono quelle del punto 1)

GOOD = otteniamo i codici dei fornitori che danno SOLO componenti ORIGINALI

R = facciamo un join con la tabella fornitore per trovare il nome dal codice fornitore

## Esercizio 2

3. Determinare le coppie di nomi di fornitori che l'intersezione dei componenti da loro forniti sia vuota.

$$\begin{aligned} F &\leftarrow \pi_{S\#, C\#}(FORNISCHE) \\ F1(S\#1, C\#1) &\leftarrow F \\ NO\_GOOD &\leftarrow \pi_{S\#, S\#1}(F \bowtie_{S\# \neq S\#1 \wedge C\# = C\#1} F1) \end{aligned}$$

$$\begin{aligned} FORN &\leftarrow \pi_{S\#}(FORNITORE) \\ FORN1(S\#1) &\leftarrow FORN \\ UNIVERSO &\leftarrow FORN \times FORN1 \end{aligned}$$

$$\begin{aligned} R\_CODICI(cod1, cod2) &\leftarrow UNIVERSO - NO\_GOOD \\ R\_NOME1(nom1, cod2) &\leftarrow \pi_{fnome, cod2}(R\_CODICI \bowtie_{cod1 = fnome} FORNITORE) \\ R\_NOMI(nom1, nom2) &\leftarrow \pi_{nom1, fnome}(R\_NOME1 \bowtie_{cod2 = fnome} FORNITORE) \end{aligned}$$

NO\_GOOD = trovo tutte le coppie di fornitori che hanno almeno una componente in comune

UNIVERSO = trovo tutte le possibili coppie di fornitori esistenti

R\_CODICI = trovo le coppie di codici di fornitori con nessuna componente in comune



4. Determinare le città in cui risiedono almeno due fornitori con status maggiore o uguale a 100, escludendo le città cui non è assegnato alcun progetto.

$$\begin{aligned} S100 &\leftarrow \pi_{S\#,citta'}(\sigma_{status \geq 100}(FORNITORE)) \\ S100\_1(S\#1, citta'1) &\leftarrow S100 \\ 2\_O\_PIU &\leftarrow \pi_{citta'}(S100 \bowtie_{S\# \neq S\#1 \wedge citta' = citta'1} S100\_1) \\ R &\leftarrow \pi_{citta'}(PROGETTO * 2\_O\_PIU) \end{aligned}$$

$S100$  = Trovo codice fornitore e città in cui lo status è maggiore o uguale a 100

$S100\_1$  = Faccio una copia del primo

$2\_O\_PIU$  = Trovo le città che hanno almeno due fornitori (ovvero codice fornitore diverso e città uguale tra le due relazioni precedenti)

$R$  = Per trovare le città che hanno almeno un progetto uso il join che mi esclude automaticamente le città senza progetti

5. Determinare i nomi dei fornitori che forniscono almeno un componente originale.

$$\begin{aligned} \text{FORNISCE1}(S\#1, C\#1) &\leftarrow \pi_{S\#, C\#}(FORNISCE) \\ \text{FORNISCE2}(S\#2, C\#2) &\leftarrow FORNISCE1 \\ \text{NON\_ORIG} &\leftarrow \pi_{C\#1}(FORNISCE1 \bowtie_{S\#1 \neq S\#2 \wedge C\#1 = C\#2} FORNISCE2) \\ \text{ORIG} &\leftarrow \pi_{C\#}(COMPONENTE) - \text{NON\_ORIG} \\ R &\leftarrow \pi_{fnome}(FORNITORE * \pi_{S\#}(FORNISCE * ORIG)) \end{aligned}$$

NON\_ORIG = Trovo le componenti non originali (= fornite da almeno due fornitori)

ORIG = Da tutte le componenti rimuovo le componenti NON originali trovando tutte le componenti originali (non so chi le offre)

R = Trovo i codici dei fornitori che forniscono le componenti originali (join naturale sul codice componente) e poi trovo il nome di questi fornitori (join naturale sul codice fornitore)

## Esercizio 2

6. Per ogni città, si determinino il massimo e il minimo dei pesi dei componenti ad esse associati.

$$\begin{aligned} COMP1(CITTA1, PESO1) &\leftarrow \pi_{citta', peso}(COMPONENTE) \\ COMP2(CITTA2, PESO2) &\leftarrow COMP1 \\ NO\_GOOD\_MAX &\leftarrow \pi_{CITTA1, PESO1}(COMP1 \bowtie_{CITTA1=CITTA2 \wedge PESO1 < PESO2} COMP2) \\ COMP\_MAX(CITTA, P\_MAX) &\leftarrow COMP1 - NO\_GOOD\_MAX \\ NO\_GOOD\_MIN &\leftarrow \pi_{CITTA1, PESO1}(COMP1 \bowtie_{CITTA1=CITTA2 \wedge PESO1 > PESO2} COMP2) \\ COMP\_MIN(CITTA, P\_MIN) &\leftarrow COMP1 - NO\_GOOD\_MIN \\ R &\leftarrow COMP\_MAX * COMP\_MIN \end{aligned}$$

Alternativa:

$$\begin{aligned} PESI\_MIN(citta, pesoMIN) &\leftarrow CITTA \mathcal{F}_{MIN(PESO)}(COMPONENTE) \\ PESI\_MAX(citta, pesoMAX) &\leftarrow CITTA \mathcal{F}_{MAX(PESO)}(COMPONENTE) \\ R &\leftarrow (PESI\_MIN * PESI\_MAX) \end{aligned}$$

7. Si determinino le città cui sono associati due o più componenti, ma non più di un progetto.

$$\begin{aligned}
 COMP1(C1, CITTA1) &\leftarrow \pi_{C\#,citta'}(COMPONENTE) \\
 COMP2(C2, CITTA2) &\leftarrow COMP1 \\
 2COMP(CITTA) &\leftarrow \pi_{CITTA1}(COMP1 \bowtie_{CITTA1=CITTA2 \wedge C1 \neq C2} COMP2) \\
 \\ 
 PROG1(P1, CITTA1) &\leftarrow \pi_{P\#,citta'}(PROGETTO) \\
 PROG2(P2, CITTA2) &\leftarrow PROG1 \\
 NO\_GOOD(CITTA) &\leftarrow \pi_{CITTA1}(PROG1 \bowtie_{CITTA1=CITTA2 \wedge P1 \neq P2} PROG2) \\
 \\ 
 R &\leftarrow 2COMP - NO\_GOOD
 \end{aligned}$$

2COMP = trovo le città con almeno due componenti

NO\_GOOD = trovo le città che hanno più di un progetto (non le vogliamo)

R = Alle città con almeno 2 componenti rimuovo quelle che hanno più di 1 progetto

## Esercizio 2

8. Si determinino i componenti (una o più) di peso massimo e quelli (una o più) di peso minimo.

$$\begin{aligned} \text{COMP\_PESI1}(C1, \text{PESO1}) &\leftarrow \pi_{C\#, \text{peso}}(\text{COMPONENTE}) \\ \text{COMP\_PESI2}(C2, \text{PESO2}) &\leftarrow \text{COMP\_PESI1} \\ \text{NO\_GOOD\_MAX} &\leftarrow \pi_{C1, \text{PESO1}}(\text{COMP\_PESI1} \bowtie_{C1 \neq C2 \wedge \text{PESO1} < \text{PESO2}} \text{COMP\_PESO2}) \\ \text{MAX} &\leftarrow \text{COMP\_PESI1} - \text{NO\_GOOD\_MAX} \\ \text{NO\_GOOD\_MIN} &\leftarrow \pi_{C1, \text{PESO1}}(\text{COMP\_PESI1} \bowtie_{C1 \neq C2 \wedge \text{PESO1} > \text{PESO2}} \text{COMP\_PESO2}) \\ \text{MIN} &\leftarrow \text{COMP\_PESI1} - \text{NO\_GOOD\_MIN} \\ R &\leftarrow \pi_{C1} \text{MAX} \cup \pi_{C1} \text{MIN} \end{aligned}$$

NO\_GOOD\_MAX = Trovo le coppie (cod componente, peso) che non hanno un peso MAX

MAX = rimuovendo le coppie precedenti trovo i componenti con peso max  
effettuo lo stesso procedimento ma inverso con quelle di peso minimo

R = unione dei codici dei componenti di peso max e di peso minimo (non ci interessa sapere il peso preciso)

Sia dato il seguente schema relazionale che descrive il calendario di una manifestazione sportiva a squadre nazionali:

*PALAZZETTO*(nome, *citta'*, *capienza*),

*INCONTRO*(nome \_\_ palazzetto, data, ora, *squadra1*, *squadra2*)

*NAZIONALE*(nazione, *continente*, *livello*).

Esprimere in algebra relazionale le seguenti interrogazioni (usando le funzioni aggregate solo se necessario e senza usare l'operatore di divisione  $\div$ ).

---

<sup>3</sup>Crediti: Prof. Angelo Montanari, corso di Basi di Dati per la Laurea Triennale, Università di Udine.

## Un esempio di possibili tabelle

nome	città	capienza
Palazzetto1	Roma	1000
Palazzetto2	Seul	2000
Palazzetto3	NY	1500
Palazzetto4	Londra	800

Table 15: Tabella PALAZZETTO

nome_palazzetto	data	ora	squadra1	squadra2
Palazzetto1	2023-06-10	20:00	Italia	Spagna
Palazzetto2	2023-06-12	18:30	Egitto	Italia
Palazzetto3	2023-06-15	21:00	Corea	India
Palazzetto4	2023-06-18	17:00	Egitto	India

Table 16: Tabella INCONTRO

nazione	continente	livello
Italia	Europa	1
Spagna	Europa	2
India	Asia	1
Corea	Asia	3
Egitto	Africa	1

Table 17: Tabella NAZIONALE

1. Determinare i nomi dei palazzetti in cui non gioca nessuna nazionale asiatica.

$$\begin{aligned}
 ASIA(naz\_asiatica) &\leftarrow \pi_{nazione}(\sigma_{continenete="Asia"}(NAZIONALE)) \\
 NO\_GOOD &\leftarrow \pi_{nome\_palazzetto}(INCONTRO \bowtie_{squadra1=naz\_asiatica \vee squadra2=naz\_asiatica} ASIA) \\
 R &\leftarrow \pi_{nome\_palazzetto}(INCONTRO) - NO\_GOOD
 \end{aligned}$$

Se assumiamo che vi possano essere palazzetti in cui non gioca alcuna squadra, la seguente soluzione è più generale:

$$R \leftarrow \pi_{nome}(PALAZZETTO) - NO\_GOOD$$

ASIA = Trovo le nazioni che si trovano nel continente 'Asia'

NO\_GOOD = Trovo i palazzetti in cui o la prima squadra è asiatica o lo è la seconda (o anche entrambe)

R = Trovo i palazzetti in cui non ci sono state squadre asiatiche



## Esercizio 3

2. Determinare la capienza complessiva dei palazzetti in cui si giocano partite di nazionali africane (ai fini della valutazione della capienza complessiva, si sommino le capienze associate a ciascuna gara, anche se più gare si svolgono nello stesso palazzetto).

$$\begin{aligned} AFRICA &\leftarrow \pi_{nazione}(\sigma_{contiente="Africa"}(NAZIONALE)) \\ INC\_AFRICA &\leftarrow \pi_{nome\_palazzetto,data,ora}(\text{INCONTRO} \bowtie_{squadra1=nazione \vee squadra2=nazione} AFRICA) \\ INC\_CAPIENZA &\leftarrow \pi_{nome\_palazzetto,data,ora,capienza}(\text{INC\_AFRICA} \\ &\quad \bowtie_{nome\_palazzetto=nome} \\ &\quad \pi_{nome,capienza}(PALAZZETTO)) \\ R &\leftarrow \mathcal{F}_{SUM(capienza)} INC\_CAPIENZA \end{aligned}$$

AFRICA = Trovo le nazioni che si trovano nel continente africa

INC\_AFRICA = Trovo i dati dei palazzetti in cui o la squadra 1 o la squadra 2 sono africane (o entrambe)

INC\_CAPIENZA = Aggiunto ai palazzetti le informazioni sulla capienza

R = Faccio la somma del totale delle capienze

## Esercizio 3

3. Determinare la città (o le città) in cui si trova il palazzetto in cui la squadra olandese gioca il maggior numero di partite.

$$\begin{aligned} INC\_OLANDA &\leftarrow \pi_{nome\_palazzetto, data, ora}(\sigma_{squadra1="Olanda" \vee squadra2="Olanda"}(INCONTRO)) \\ NUM\_INC1(P1, NUM1) &\leftarrow nome\_palazzetto \mathcal{F}_{COUNT}(data, ora) INC\_OLANDA \\ NUM\_INC2(P2, NUM2) &\leftarrow NUM\_INC1 \\ NO\_GOOD &\leftarrow \pi_{P1}(NUM\_INC1 \bowtie_{P1 \neq P2 \wedge NUM1 < NUM2} NUM\_INC2) \\ GOOD &\leftarrow \pi_{P1}(NUM\_INC1) - NO\_GOOD \\ R &\leftarrow \pi_{citta'}(GOOD \bowtie_{P1=nome} PALAZZETTO) \end{aligned}$$

Si noti che  $I\_OL$  si può calcolare alternativamente come:

$$\begin{aligned} INC\_OLANDA &\leftarrow \pi_{nome\_palazzetto, data, ora}(\sigma_{squadra1="Olanda"}(INCONTRO)) \\ &\cup \\ &\pi_{nome\_palazzetto, data, ora}(\sigma_{squadra2="Olanda"}(INCONTRO)) \end{aligned}$$

INC\_OLANDA = Seleziono gli incontri svolti dalla squadra Olanda

NUM\_INC1 = Calcolo il numero di partite svolte dall'olanda per ogni palazzetto

NO\_GOOD = Trovo i palazzetti con numero di partire NON massimo

GOOD = Trovo il palazzetto con numero di partite max

R = Trovo l'informazione sulla città del palazzetto

## Esercizio 3

4. Determinare le squadre che incontrano solo squadre dello stesso livello.

$$\begin{aligned}RIS1(SQUAD1, SQUAD2, LVL1) &\leftarrow \pi_{squadra1, squadra2, livello}(\text{INCONTRO} \bowtie_{squadra1=nazione} NAZIONALE) \\RIS2(SQUAD1, SQUAD2, LVL1, LVL2) &\leftarrow \pi_{SQUAD1, SQUAD2, LVL1, livello}(\text{R1} \bowtie_{S2=nazione} NAZIONALE) \\NO\_GOOD1 &\leftarrow \pi_{SQUAD1}(\sigma_{LVL1 \neq LVL2} (RIS2)) \\NO\_GOOD2 &\leftarrow \pi_{SQUAD2}(\sigma_{LVL1 \neq LVL2} (RIS2)) \\NO\_GOOD &\leftarrow NO\_GOOD1 \cup NO\_GOOD2 \\R &\leftarrow \pi_{nazione}(NAZIONALE) - NO\_GOOD\end{aligned}$$

Il ragionamento procede all'opposto: trovo le squadre che fanno incontri con squadre di altri livelli e le tolgo dal totale.

RIS1 = Trovo le informazioni sul livello della squadra1 (join con nazionale)

RIS2 = Trovo le informazioni sul livello della squadra2

NO\_GOOD1 = Trovo quali sono le squad1 che sono andate contro squadre2 di altri livelli

NO\_GOOD2 = Trovo quali sono le squad2 che sono andate contro squadrel di altri livelli

NO\_GOOD = Faccio un'unione delle squadre che vanno contro altri livelli

R = Rimuovo dall'elenco di tutte le squadre quelle che non ci interessano.