

RA: Un interprete per l'Algebra Relazionale

Gianluca Cima, Maurizio Lenzerini



SAPIENZA
UNIVERSITÀ DI ROMA

Anno accademico 2021/2022

Come procurarsi il sistema RA

In generale, l'interprete dell'algebra relazionale RA si può scaricare seguendo il seguente link, dove si trova anche la documentazione del software:

<https://users.cs.duke.edu/~junyang/ra2/>

Dal sito raggiungibile con il link sopra specificato si deve scaricare il file `ra.jar` (che è proprio il programma corrispondente all'interprete per l'algebra relazionale usato in laboratorio) ed il file *sample.properties*, necessario per il funzionamento del software.

Come utilizzare RA

1. Per utilizzare RA bisogna preparare in una directory i seguenti file:
 - *ra.jar*,
 - *sample.properties*,
 - un file con estensione *.ra*, che contiene uno script che va a creare una base di dati, con lo schema delle relazioni e le loro istanze (dal sito di cui sopra si può scaricare un file con estensione *.ra* di esempio).

In particolare, il file *sample.properties* serve a specificare il nome del file che conterrà tutte le informazioni sulla base di dati che vogliamo usare. Ad esempio, per utilizzare una base di dati di nome “alfa”, il file *sample.properties* deve contenere la linea

`url=jdbc:sqlite:alfa.db`

perché questo è il meccanismo per assegnare il nome alla base di dati che vogliamo usare.

2. Dirigiamoci tramite la shell del sistema operativo nella cartella dove sono presenti i tre file menzionati sopra, e diamo il seguente comando:

```
java -jar ra.jar sample.properties -i nome-file.ra
```

Questo comando legge il file *nome-file.ra* e crea il file *nome.db*, dove “*nome*” è il nome che abbiamo specificato nel file *sample.properties*.

3. A questo punto il file *nome.db* contiene tutte le informazioni sulla base di dati e la seguente istruzione permette di utilizzare l'interprete RA usando la base di dati appena creata:

```
java -jar ra.jar sample.properties
```

Per questa esercitazione

Nella sezione “Esercitazioni” della pagina Web del corso <http://www.diag.uniroma1.it/~lenzerini/index.html/?q=node/44> abbiamo preparato tutto il necessario:

- il file *slides di guida sull'eseritazione* (queste slides),
- il file *ra.jar*,
- il file *sample.properties*,
- il file *birre.ra* (la base di dati per gli esercizi in formato ra),
- il file *basedidati.txt* (la base di dati per gli esercizi in formato testo),
- il file *queries.pdf* (con la specifica delle query da scrivere).

Per svolgere l'esercitazione occorre scaricare i file suddetti, posizionarli in una nuova cartella sul Desktop, dirigerci tramite la shell nella cartella e poi creare il file “birre.db” tramite il comando:

```
java -jar ra.jar sample.properties -i birre.ra
```

Si può iniziare ad utilizzare l'interprete RA con la base di dati appena creata tramite il comando:

```
java -jar ra.jar sample.properties
```

L'esercitazione consiste nello scrivere le query proposte in “queries.pdf”.

D'ora in poi assumeremo di essere all'interno dell'interprete RA (**stare attenti al punto e virgola alla fine dei comandi!**).

- Il seguente comando produce l'elenco di tutte le relazioni presenti nel DB:

`\list;`

- Per avere l'elenco delle tuple della relazione di nome *relName* occorre lanciare il comando:

`relName;`

- Per uscire dal sistema occorre lanciare il comando:

`\quit;`

Nel seguito, illustriamo i comandi necessari per usare in RA gli operatori dell'algebra relazionale.

Unione - comando `\union`

Assumiamo di avere le seguenti due relazioni nella nostra base di dati (il dominio *text* equivale a *string*):

- Laureato(matricola: *text*, nome:*text*)
 $\{\langle 7274, Rossi \rangle, \langle 7432, Neri \rangle\}$
- Quadro(matricola: *text*, nome: *text*)
 $\{\langle 7432, Neri \rangle, \langle 9297, Neri \rangle\}$

Vogliamo calcolare l'unione delle due relazioni:

- Algebra relazionale sulle slide: Laureato \cup Quadro
- **In RA:** Laureato `\union` Quadro;
- Risultato: schema (matricola: *text*, nome: *text*)
 $\{\langle 7274, Rossi \rangle, \langle 7432, Neri \rangle, \langle 9297, Neri \rangle\}$

Intersezione - comando `\intersect`

Assumiamo di avere le seguenti due relazioni:

- Laureato(matricola: *text*, nome: *text*)
 $\{\langle 7274, Rossi \rangle, \langle 7432, Neri \rangle\}$
- Quadro(matricola: *text*, nome: *text*)
 $\{\langle 7432, Neri \rangle, \langle 9297, Neri \rangle\}$

Vogliamo calcolare l'intersezione delle due relazioni:

- Algebra relazionale sulle slide: Laureato \cap Quadro
- **In RA:** Laureato `\intersect` Quadro;
- Risultato: schema (matricola: *text*, nome: *text*)
 $\{\langle 7432, Neri \rangle\}$

Assumiamo di avere le seguenti due relazioni:

- Laureato(matricola: *text*, nome: *text*)
 $\{\langle 7274, Rossi \rangle, \langle 7432, Neri \rangle\}$
- Quadro(matricola: *text*, nome: *text*)
 $\{\langle 7432, Neri \rangle, \langle 9297, Neri \rangle\}$

Vogliamo calcolare la differenza tra le due relazioni:

- Algebra relazionale sulle slide: Laureato – Quadro
- **In RA:** Laureato `\diff` Quadro;
- Risultato: schema (matricola: *text*, nome: *text*)
 $\{\langle 7274, Rossi \rangle\}$

In RA la ridenominazione è **diversa da quanto abbiamo visto a lezione**: essa va fatta inserendo la lista completa dei nuovi nomi (uno per ciascun attributo dell'operando) per tutti gli attributi, separati da virgola, .

Assumiamo di avere la seguente relazione:

- Paternità(padre: *text*, figlio : *text*)
 $\{\langle \textit{Adamo}, \textit{Abele} \rangle, \langle \textit{Adamo}, \textit{Caino} \rangle, \langle \textit{Abramo}, \textit{Isacco} \rangle\}$

Vogliamo calcolare la relazione ottenuta ridenominando padre in genitore:

- Algebra relazionale sulle slide: $\text{REN}_{\text{genitore} \leftarrow \text{padre}}(\text{Paternità})$
- **In RA**: $\backslash \text{rename}_{\{\text{genitore}, \text{figlio}\}}(\text{Paternità})$;
- Risultato: schema (genitore: *text*, figlio: *text*)
 $\{\langle \textit{Adamo}, \textit{Abele} \rangle, \langle \textit{Adamo}, \textit{Caino} \rangle, \langle \textit{Abramo}, \textit{Isacco} \rangle\}$

Esempio di utilizzo della ridenominazione

Assumiamo di avere le seguenti due relazioni:

- Paternità(padre: *text*, figlio: *text*)
 $\{\langle \text{Adamo}, \text{Abele} \rangle, \langle \text{Adamo}, \text{Caino} \rangle, \langle \text{Abramo}, \text{Isacco} \rangle\}$
- Maternità(madre: *text*, figlio: *text*)
 $\{\langle \text{Eva}, \text{Abele} \rangle, \langle \text{Eva}, \text{Set} \rangle, \langle \text{Sara}, \text{Isacco} \rangle\}$

Vogliamo calcolare la loro unione:

- Algebra relazionale sulle slide: $\text{REN}_{\text{genitore} \leftarrow \text{padre}}(\text{Paternità}) \cup \text{REN}_{\text{genitore} \leftarrow \text{madre}}(\text{Maternità})$
- In RA: $\backslash \text{rename_}\{\text{genitore}, \text{figlio}\}(\text{Paternità}) \backslash \text{union} \backslash \text{rename_}\{\text{genitore}, \text{figlio}\}(\text{Maternità});$
- Risultato: schema (genitore: *text*, figlio: *text*)
 $\{\langle \text{Adamo}, \text{Abele} \rangle, \langle \text{Adamo}, \text{Caino} \rangle, \langle \text{Abramo}, \text{Isacco} \rangle, \langle \text{Eva}, \text{Abele} \rangle, \langle \text{Eva}, \text{Set} \rangle, \langle \text{Sara}, \text{Isacco} \rangle\}$

Assumiamo di avere la seguente relazione:

- Impiegato(matricola: *text*, cognome: *text*, filiale *text*, stipendio *integer*)
 $\{\langle 7309, Rossi, Roma, 55 \rangle, \langle 5998, Neri, Milano, 64 \rangle, \langle 9553, Milano, Milano, 44 \rangle, \langle 5698, Neri, Napoli, 64 \rangle\}$

Vogliamo gli impiegati che guadagnano più di 50 e che lavorano a Milano o a Roma:

- Algebra relazionale sulle slide:
 $SEL_{\text{stipendio} > 50 \text{ AND (filiale='Milano' OR filiale='Roma')}}(\text{Impiegato})$
- In RA: \select_{\text{stipendio} > 50 \text{ AND (filiale='Milano' OR filiale='Roma')}}(\text{Impiegato});
- Risultato: schema (matricola: *text*, cognome: *text*, filiale *text*, stipendio *integer*)
 $\{\langle 7309, Rossi, Roma, 55 \rangle, \langle 5998, Neri, Milano, 64 \rangle\}$

Assumiamo di avere la seguente relazione:

- $\text{Impiegato}(\text{matricola: text, cognome: text, filiale: text, stipendio: integer})$
 $\{\langle 7309, \text{Neri}, \text{Napoli}, 55 \rangle, \langle 5998, \text{Neri}, \text{Milano}, 64 \rangle,$
 $\langle 9553, \text{Rossi}, \text{Roma}, 44 \rangle, \langle 5698, \text{Rossi}, \text{Roma}, 64 \rangle\}$

Vogliamo il cognome e la filiale di tutti gli impiegati:

- Algebra relazionale sulle slide: $\text{PROJ}_{\text{Cognome, Filiale}}(\text{Impiegato})$
- In RA: $\backslash\text{project}_{\{\text{Cognome, Filiale}\}}(\text{Impiegato});$
- Risultato: schema (cognome: text, filiale: text)
 $\{\langle \text{Neri}, \text{Napoli} \rangle, \langle \text{Neri}, \text{Milano} \rangle, \langle \text{Rossi}, \text{Roma} \rangle\}$

Join naturale - comando `\join`

Assumiamo di avere le seguenti due relazioni:

- Docente(codice: *text*, corso: *text*)
 $\{\langle 1, BD \rangle, \langle 2, PS \rangle, \langle 3, Reti \rangle, \langle 1, PS \rangle\}$
- Silnsegna(corso: *text*, ling: *text*)
 $\{\langle BD, SQL \rangle, \langle BD, JAVA \rangle, \langle PS, UML \rangle, \langle KRST, DATALOG \rangle\}$

Vogliamo sapere i docenti quali corsi e quali linguaggi insegnano:

- Algebra relazionale sulle slide: Docente JOIN Silnsegna
- **In RA:** Docente `\join` Silnsegna;
- Risultato: schema (codice: *text*, corso: *text*, ling: *text*)
 $\{\langle 1, BD, SQL \rangle, \langle 1, BD, JAVA \rangle, \langle 2, PS, UML \rangle, \langle 1, PS, UML \rangle\}$

Prodotto cartesiano - comando `\cross`

Assumiamo di avere le seguenti due relazioni:

- `Impiegato`(`impiegato: text`, `reparto: text`)
 $\{\langle Rossi, A \rangle, \langle Neri, B \rangle, \langle Bianchi, B \rangle\}$
- `Reparto`(`codice: text`, `capo: text`)
 $\{\langle A, Mori \rangle, \langle B, Bruni \rangle\}$

Vogliamo accoppiare tutti gli impiegati con tutti i reparti:

- Algebra relazionale sulle slide: `Impiegato JOIN Reparto`
- **In RA:** `Codice \cross Reparto` (o, in questo caso, equivalentemente `Codice \join Reparto`);
- Risultato: schema (`impiegato: text`, `reparto: text`, `codice: text`, `capo: text`)
 $\{\langle Rossi, A, A, Mori \rangle, \langle Rossi, A, B, Bruni \rangle, \langle Neri, B, A, Mori \rangle, \langle Neri, B, B, Bruni \rangle, \langle Bianchi, B, A, Mori \rangle, \langle Bianchi, B, B, Bruni \rangle\}$

Theta-join - comando `\join` con condizione

Assumiamo di avere le seguenti due relazioni:

- `Impiegato`(`impiegato: text`, `reparto: text`)
 $\{\langle Rossi, A \rangle, \langle Neri, B \rangle, \langle Bianchi, B \rangle\}$
- `Reparto`(`codice: text`, `capo: text`)
 $\{\langle A, Mori \rangle, \langle B, Bruni \rangle\}$

Vogliamo sapere il reparto ed il capo di ogni impiegato:

- Algebra relazionale sulle slide: `Impiegato JOINreparto=codice Reparto`
- **In RA:** `Impiegato \join_{reparto=codice} Reparto;`
- Risultato: schema (`impiegato: text`, `reparto: text`, `codice: text`, `capo: text`)
 $\{\langle Rossi, A, A, Mori \rangle, \langle Neri, B, B, Bruni \rangle, \langle Bianchi, B, B, Bruni \rangle\}$

Lo schema della base di dati è:

```
Bar(nome:string, indirizzo:string)
Bevitore(nome:string, indirizzo:string)
Birra(nome:string, fabbrica:string)
Frequenta(bevitore:string, bar:string, volte_a_settimana:number)
Gradisce(bevitore:string, birra:string)
Serve(bar:string, birra:string, prezzo:number)
```

Di seguito, la specifica delle query e le corrispondenti soluzioni.

1. Calcolare il nome dei bar che servono la birra Peroni
2. Calcolare il nome dei bevitori che gradiscono la birra Amstel o la birra Corona
3. Calcolare il nome e l'indirizzo dei bar in cui viene servita almeno una birra con prezzo maggiore di 2
4. Calcolare nome e indirizzo dei bevitori che frequentano almeno un bar almeno due volte a settimana
5. Calcolare nome e indirizzo dei bevitori che frequentano almeno un bar che serve la birra Peroni
6. Calcolare le coppie di bevitori che risiedono nello stesso indirizzo
7. Calcolare le coppie di bevitori che risiedono nello stesso indirizzo, evitando nel risultato sia ogni coppia $\langle x, x \rangle$, sia ogni coppia $\langle x, y \rangle$ se già compare $\langle y, x \rangle$
8. Calcolare quali bevitori gradiscono sia la birra Amstel sia la birra Corona
9. Calcolare il nome dei bevitori che frequentano almeno due bar diversi
10. Calcolare le birre che non sono servite da alcun bar