

I concetti del Modello Relazionale - Parte 3

Prof. Francesco Gobbi

I.I.S.S. Galileo Galilei, Ostiglia

28 novembre 2024

Concetti chiave: Insiemi

- ▶ Un insieme è una raccolta di elementi distinti.
- ▶ Esempio:

$$A = \{1, 2, 3\}, \quad B = \{3, 4, 5\}$$

- ▶ Operazioni fondamentali:
 - ▶ **Unione** ($A \cup B$): Combina tutti gli elementi di A e B :

$$A \cup B = \{1, 2, 3, 4, 5\}$$

- ▶ **Intersezione** ($A \cap B$): Elementi comuni a A e B :

$$A \cap B = \{3\}$$

- ▶ **Differenza** ($A - B$): Elementi in A ma non in B :

$$A - B = \{1, 2\}$$

Che cos'è l'Algebra Relazionale?

► Linguaggio formale:

- L'Algebra Relazionale è un linguaggio formale utilizzato per manipolare e interrogare dati organizzati secondo il **modello relazionale**.
- Ogni operazione nell'Algebra Relazionale prende in input una o più **relazioni** (tabelle) e produce un'altra relazione come output.

► Obiettivo:

- Fornire un insieme di operatori per descrivere **cosa vogliamo ottenere dai dati**, senza specificare come i dati debbano essere elaborati (approccio dichiarativo).
- È alla base di molti linguaggi di interrogazione come SQL o MySQL, che implementano concetti dell'Algebra Relazionale.

Che cos'è l'Algebra Relazionale?

► Struttura di base:

► Relazioni:

- Una relazione è rappresentata come una **tabella**, costituita da:
- **Righe** (*tuple*): Ogni riga rappresenta un record o un'istanza del dato.
- **Colonne** (*attributi*): Ogni colonna rappresenta una proprietà o un campo del record.
- Le relazioni sono insiemi: non contengono duplicati e l'ordine delle righe non è significativo.

► Operatori:

- Gli operatori nell'Algebra Relazionale consentono di:
- **Selezionare**: Filtrare le righe in base a condizioni (es. età > 18).
- **Proiettare**: Ridurre la relazione alle sole colonne di interesse.
- **Unire**: Combinare due relazioni in base a una condizione comune.
- **Effettuare operazioni insiemistiche**: Unione, intersezione, differenza.

Esempio: Tabelle "Studenti" e "Corsi"

Tabella "Studenti"

ID_Studente	Nome	Cognome	Corso
1	Marco	Rossi	Informatica
2	Laura	Bianchi	Matematica
3	Sara	Verdi	Informatica

Tabella "Corsi"

ID_Corso	Nome_Corso	Docente
101	Informatica	Prof. Neri
102	Matematica	Prof. ssa Bianchi

Domanda Iniziale da porsi

- ▶ Come possiamo estrarre **tutti gli studenti che frequentano il corso di Informatica?**
- ▶ Utilizziamo l'**operatore di selezione** (σ):

$$\sigma_{\text{Corso} = \text{'Informatica'}}(\text{Studenti})$$

- ▶ Risultato:

ID_Studente	Nome	Cognome	Corso
1	Marco	Rossi	Informatica
3	Sara	Verdi	Informatica

1. Operatore di Selezione (σ)

- Filtra le tuple(righe) di una relazione in base a una condizione specificata e crea una nuova tabella con solo queste tuple estratte.

- **Notazione:**

$$\sigma_{\text{condizione}}(\text{Relazione})$$

- **Esempio:** Trova tutti gli studenti che frequentano Informatica.

$$\sigma_{\text{Corso} = \text{'Informatica'}}(\text{Studenti})$$

- **Risultato:**

ID_Studente	Nome	Cognome	Corso
1	Marco	Rossi	Informatica
3	Sara	Verdi	Informatica

2. Operatore di Proiezione (π)

- Seleziona specifici attributi (colonne) di una relazione e genera una nuova tabella con solo questi nuovi attributi estratti.

- **Notazione:**

$$\pi_{\text{attributi}}(\text{Relazione})$$

- **Esempio:** Visualizza nome e cognome degli studenti.

$$\pi_{\text{Nome, Cognome}}(\text{Studenti})$$

- **Risultato:**

Nome	Cognome
Marco	Rossi
Laura	Bianchi
Sara	Verdi

3. Operatore di Unione (\cup)

- ▶ Combina due relazioni con lo stesso schema in una sola tabella nuova generata, includendo tutte le righe presenti in almeno una delle due.
- ▶ **Notazione:**

$$R \cup S$$

- ▶ **Esempio:** Studenti di Informatica e Matematica.
- ▶ Tabelle di partenza:

R =	ID_Studente	Nome	S =	ID_Studente	Nome
	1	Marco		2	Laura
	3	Sara		3	Sara

- ▶ **Risultato:**

ID_Studente	Nome
1	Marco
2	Laura
3	Sara

4. Operatori di Intersezione (\cap) e Differenza ($-$)

- ▶ **Intersezione (\cap):** Restituisce le righe comuni a due relazioni.

$$R \cap S$$

- ▶ **Differenza ($-$):** Restituisce le righe presenti in una relazione ma non nell'altra.

$$R - S$$

- ▶ **Esempio:** Studenti iscritti a Informatica ma non a Matematica.

5. Operatore di Join (\bowtie)

- ▶ Un join combina due relazioni (tabelle) basandosi su una condizione specifica creando una nuova relazione.
- ▶ **Notazione:**

$$R \bowtie_{\text{condizione}} S$$

Tipologie principali:

- ▶ 5.1 **Inner Join** (Equi e Naturale)
- ▶ 5.2 **Outer Join** (Left, Right, Full)

5.1.1 Equi Join

Definizione: Combina le righe di due relazioni basandosi su una condizione di uguaglianza esplicita tra attributi.

Notazione:

$$R \bowtie_{A = B} S$$

Esempio:

Tabella Studenti

ID	Nome
1	Marco
2	Laura
3	Sara

Tabella Corsi

ID_Corso	Docente
1	Prof. Neri
3	Prof. ssa Verdi

Operazione:

$$\text{Studenti} \bowtie_{\text{Studenti.ID} = \text{Corsi.ID_Corso}} \text{Corsi}$$

Risultato:

ID	Nome	ID_Corso	Docente
1	Marco	1	Prof. Neri
3	Sara	3	Prof. ssa Verdi

5.1.1 Equi Join

Considerazioni:

Grado della relazione risultante: La somma degli attributi delle due tabelle iniziali.

$$\text{Grado: } \text{Grado}(\text{Studenti}) + \text{Grado}(\text{Corsi}) = 2 + 2 = 4$$

Cardinalità della relazione risultante: Dipende dalla condizione; nel caso di corrispondenza, il numero massimo di righe è determinato dalla tabella con meno righe.

Cardinalità: 2 (corrisponde alle righe unite correttamente.)

L'Equi Join viene utilizzato in esercizi in cui è **richiesto il confronto tra attributi, anche senza lo stesso nome**. È quindi più versatile nelle varie richieste ed in accoppiata con selezione e proiezione può comunque dare un grado finale della relazione minore.

5.1.2 Join Naturale

Definizione: Combina le righe di due relazioni basandosi implicitamente su tutti gli attributi comuni con lo stesso nome. Gli attributi comuni compaiono una sola volta nel risultato.

Notazione:

$$R \bowtie S$$

Esempio: Tabelle:

Studenti	ID	Nome
	1	Marco
	2	Laura
	3	Sara

Corsi	ID	Docente
	1	Prof. Neri
	3	Prof. ssaVerdi

Operazione:

$$\text{Studenti} \bowtie \text{Corsi}$$

Risultato:

ID	Nome	Docente
1	Marco	Prof. Neri
3	Sara	Prof. ssaVerdi

5.1.2 Join Naturale

Considerazioni:

Grado della relazione risultante: La somma degli attributi delle due tabelle meno quelli comuni (mostrati una sola volta).

Grado: $\text{Grado}(\text{Studenti}) + \text{Grado}(\text{Corsi}) - \text{Attributi comuni} = 2 + 2 - 1 = 3$

Cardinalità della relazione risultante: Dipende dalla condizione implicita; il numero massimo di righe è determinato dalla tabella con meno righe.

Cardinalità: 2 (corrisponde alle righe unite correttamente.)

5.2 Outer Join (Left, Right, Full)

- ▶ Include tutte le righe di una tabella (o di entrambe) anche se non hanno corrispondenza nell'altra tabella.
- ▶ **Tipologie:**
- ▶ **Left Outer Join:** Include tutte le righe della tabella di sinistra.
- ▶ **Right Outer Join:** Include tutte le righe della tabella di destra.
- ▶ **Full Outer Join:** Include tutte le righe di entrambe le tabelle.
- ▶ **Esempio (Left Join):**

Studenti LEFT JOIN Corsi ON Studenti.ID = Corsi.ID_Corso

- ▶ **Risultato:**

Nome	ID_Corso	Docente
Marco	1	Prof.Neri
Laura	NULL	NULL
Sara	3	Prof.ssaVerdi

Esempi di Outer Join: Tabelle di riferimento

Tabella Dipendenti

ID_Dip	Nome	Cognome
1	Marco	Rossi
2	Laura	Bianchi
3	Sara	Verdi

Tabella Progetti

ID_Progetto	Nome_Progetto	ID_Dip
101	ProgettoAlpha	1
102	ProgettoBeta	2
103	ProgettoGamma	NULL

5.2.1 Left Outer Join

Descrizione: Include tutte le righe della tabella di sinistra (**Dipendenti**), anche se non trovano corrispondenza nella tabella di destra (**Progetti**). **Operazione:**

Dipendenti LEFT JOIN Progetti ON

Dipendenti.ID_Dip = Progetti.ID_Dip

Risultato:

ID_Dip	Nome	Cognome	Nome_Progetto	ID_Progetto
1	Marco	Rossi	ProgettoAlpha	101
2	Laura	Bianchi	ProgettoBeta	102
3	Sara	Verdi	NULL	NULL

5.2.2 Right Outer Join

Descrizione: Include tutte le righe della tabella di destra (**Progetti**), anche se non trovano corrispondenza nella tabella di sinistra (**Dipendenti**). **Operazione:**

Dipendenti RIGHT JOIN Progetti ON

Dipendenti.ID_Dip = Progetti.ID_Dip

Risultato:

ID_Dip	Nome	Cognome	Nome_Progetto	ID_Progetto
1	Marco	Rossi	ProgettoAlpha	101
2	Laura	Bianchi	ProgettoBeta	102
NULL	NULL	NULL	ProgettoGamma	103

5.2.3 Full Outer Join

Descrizione: Combina Left e Right Outer Join, includendo tutte le righe di entrambe le tabelle. **Operazione:**

Dipendenti FULL OUTER JOIN Progetti ON

Dipendenti.ID_Dip = Progetti.ID_Dip

Risultato:

ID_Dip	Nome	Cognome	Nome_Progetto	ID_Progetto
1	Marco	Rossi	ProgettoAlpha	101
2	Laura	Bianchi	ProgettoBeta	102
3	Sara	Verdi	NULL	NULL
NULL	NULL	NULL	ProgettoGamma	103