

# Basi di Dati 1

Simone Lidonnici

23 aprile 2024

# Indice

<b>1</b>	<b>Modello relazionale</b>	<b>2</b>
1.1	Relazioni come tabelle . . . . .	2
1.2	Scrivere tabelle corrette . . . . .	3
1.2.1	Vincoli di integrità . . . . .	4
1.2.2	Chiavi primarie . . . . .	4
1.3	Dipendenze funzionali . . . . .	5
<b>2</b>	<b>Algebra relazionale</b>	<b>6</b>
2.1	Basi dell'algebra relazionale . . . . .	6
2.2	Modifica di una relazione . . . . .	6
2.2.1	Proiezione . . . . .	6
2.2.2	Selezione . . . . .	7
2.2.3	Rinomina . . . . .	8
2.3	Operazioni insiemistiche . . . . .	8
2.3.1	Unione . . . . .	8
2.3.2	Differenza . . . . .	9
2.3.3	Intersezione . . . . .	9
2.4	Operazioni di combinazione . . . . .	10
2.4.1	Prodotto cartesiano . . . . .	10
2.4.2	Concatenazione . . . . .	10

# 1

## Modello relazionale

### Prodotto cartesiano tra domini

Il **dominio** è un insieme, anche infinito, di valori utilizzabili.

Presi  $D_1, D_2, \dots, D_k$  domini, non necessariamente diversi, il **prodotto cartesiano**, scritto:

$$D_1 \times D_2 \times \dots \times D_k$$

é l'insieme:

$$\{(v_1, v_2, \dots, v_k) | v_i \in D_i \forall i\}$$

#### Esempio:

$$D_1 = \{a, b\}$$

$$D_2 = \{x, y, z\}$$

$$D_1 \times D_2 = \{(a, x), (a, y), (a, z), (b, x), (b, y), (b, z)\}$$

### Relazione

Una **relazione**  $r$  è un qualsiasi sottoinsieme di un prodotto cartesiano:

$$r \subseteq D_1 \times D_2 \times \dots \times D_k$$

Il **grado** della relazione è pari al valore di  $k$ , cioè al numero dei domini.

In una relazione, ogni elemento  $t \in r$  è una **tupla** che contiene numero di elementi pari al grado della relazione.

L'elemento  $i$ -esimo di una tupla viene indicato come  $t[i]$  oppure  $t.i$  con  $i \in [1, k]$ .

Una relazione ha una **cardinalità** pari al numero di tuple che la compongono.

#### Esempio:

$$D_1 = \{\text{white}, \text{black}\}$$

$$D_2 = \{0, 1, 2\}$$

$$D_1 \times D_2 = \{(\text{white}, 0), (\text{white}, 1), (\text{white}, 2), (\text{black}, 0), (\text{black}, 1), (\text{black}, 2)\}$$

La relazione  $r = \{(\text{white}, 0), (\text{black}, 1), (\text{black}, 2)\}$  è una relazione di grado 2 e cardinalità 3

La tupla  $t = (\text{white}, 0) \implies t[1] = \text{white}$  e  $t[2] = 0$

## 1.1 Relazioni come tabelle

Le relazioni possono essere rappresentate come tabelle, in cui viene dato un nome ad ogni colonna.

### Attributi e schema relazionale

Un **attributo**  $A$  è una coppia  $(A, dom(A))$  in cui  $A$  è il nome dell'attributo e  $dom(A)$  il suo dominio. Due attributi possono avere stesso dominio ma non stesso nome.

Uno **schema relazionale** è l'insieme di tutti gli attributi che definiscono la relazione associata allo schema stesso:

$$R(A_1, A_2, \dots, A_k)$$

In uno schema relazionale, una tupla è una funzione che associa ad ogni attributo  $A$  un elemento del  $dom(A)$ . Con  $t[A]$  indichiamo il valore della tupla corrispondente all'attributo  $A$ .

Un insieme di tuple  $r$  è un'**istanza di una relazione**.

**Esempio:**

$R = \{(Nome, String), (Cognome, String), (Media, Real)\}$

Nome	Cognome	Media
Marco	Casu	28
Simone	Lidonnici	27

### Schema di un database

Un insieme di relazioni distinte è uno **schema di un database**:

$$(R_1, R_2, \dots, R_n)$$

Un database relazionale è uno schema di un database in cui sono definite le istanze  $(r_1, r_2, \dots, r_n)$  in cui  $r_i$  è un'istanza di  $R_i \forall i$ .

## 1.2 Scrivere tabelle corrette

I collegamenti tra i vari dati in schemi diversi vengono fatti attraverso i valori.

**Esempio:**

Matricola	Nome	Cognome
2041612	Marco	Casu
2061343	Simone	Lidonnici

Matricola	Voto	Esame
2041612	30	Sistemi
2061343	28	Algoritmi

In alcuni casi i potrebbero mancare delle informazioni, in questi casi si inserisce **NULL** come valore nella tupla, nel campo dell'attributo mancante. **NULL** è un valore che non appartiene a nessun dominio ma può rimpiazzare qualunque valore.

**Esempio:**

Matricola	Nome	Cognome
NULL	Nadia	Ge
2061343	NULL	Lidonnici

### 1.2.1 Vincoli di integrità

I dati devono avere dei vincoli da rispettare, questi vincoli possono essere:

- **Vincolo di dominio:** valori che devono essere per forza contenuti in un insieme specifico
- **Vincolo intra-relazionale:** sono vincoli tra valori di una stessa tupla
- **Vincolo inter-relazionale:** sono vincoli tra valori di diverse istanze
- **Vincolo di chiave primaria:** il valore deve essere diverso da NULL e unico negli attributi appartenenti alla chiave
- **Vincolo di esistenza:** il valore deve per forza essere diversa da NULL

**Esempio:**

Studenti:

Matricola	Nome	Cognome
2041612	Marco	Casu
2061343	Simone	Lidonnici

Voti:

Matricola	Esame	Voto	Lode
2061343	Probabilità	30	Si
2041612	Sistemi	28	NULL

In questo caso i vincoli sono:

- Vincolo di dominio:  $18 \leq \text{Voto} \leq 30$
- Vincolo intra-relazionale:  $\neg(\text{Voto} = 30) \implies \neg(\text{Lode} = \text{Si})$
- Vincolo inter-relazione: La matricola nello schema Voti deve esistere nello schema Studenti
- Vincolo di chiave primaria: La matricola nello schema Studenti deve per forza esistere ed essere unica
- Vincolo di esistenza: Il nome nello schema Studenti deve per forza esistere

### 1.2.2 Chiavi primarie

#### Chiave di una relazione

Una **chiave** di una relazione è un attributo o un gruppo di attributi che devono esistere ed appartenere ad una sola tupla.

Un insieme  $X$  di attributi dello schema  $R$  è una chiave di  $R$  se:

1. Per ogni istanza di  $R$  non esistono due tuple con gli stessi valori in tutti gli attributi di  $X$ , cioè  $\forall t_1, t_2 \in r \quad t_1[X] = t_2[X] \implies t_1 = t_2$
2. Per ogni sottoinsieme  $X' \subset X$  possono esserci delle tuple che hanno tutti gli attributi di  $X'$  uguali ma sono diverse, cioè  $\forall X' \subset X \quad \exists t_1, t_2 | t_1[X'] = t_2[X'] \text{ ma } t_1 \neq t_2$

**Esempio:**

Lezione	Edificio	Aula	Orario
Basi di Dati	Chimica	2	16:00
Algebra	Informatica	2	17:00
Probabilità	Matematica	4	15:30

$X = \{\text{Edificio, Aula, Orario}\}$  è una chiave primaria perché due lezioni non possono essere nello stesso edificio, aula e ora.

Togliendo però qualsiasi dei tre si possono avere delle lezioni diverse che abbiano gli altri attributi uguali (lezioni nello stesso edificio e aula ma ore diverse, nella stessa aula e ora ma edificio diverso o nello stesso edificio e ora ma aule diverse).

## 1.3 Dipendenze funzionali

### Dipendenza funzionale

Una **dipendenza funzionale** è un insieme di attributi:

$$X, Y \subseteq R \mid X \rightarrow Y$$

Si dice “X determina Y” in cui  $X$  è il determinante e  $Y$  è il dipendente.

### Esempio:

Flight(Code,Day,Pilot,Time)

Questo schema ha dei vincoli dati dal buonsenso:

- un volo con un codice parte sempre alla stessa
- un volo parte con un solo pilota, in un determinato orario e tempo

Questi vincoli determinano delle dipendenze funzionali:

- $\text{Code} \rightarrow \text{Time}$
- $\{\text{Day, Pilot, Time}\} \rightarrow \text{Code}$
- $\{\text{Code, Day}\} \rightarrow \text{Pilot}$

### Soddisfazione di una dipendenza funzionale

Un'istanza  $r$  di uno schema  $R$  soddisfa la dipendenza funzionale  $X \rightarrow Y$  se:

1. La dipendenza è applicabile su  $R$ , cioè  $X \subset R \wedge Y \subset R$
2. Due tuple in  $R$  che hanno gli stessi attributi  $X$  avranno anche gli stessi attributi  $Y$ , cioè  $\forall t_1, t_2 \in r \ t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$

# 2

## Algebra relazionale

L'algebra relazionale è una notazione per specifiche domande (dette query) sul contenuto delle relazioni, le operazioni in algebra relazionale hanno una controparte in SQL. Per processare una query il DBMS trasforma SQL in una notazione simile all'algebra relazionale.

Il linguaggio per interrogare un database è un insieme di operatori binari o unitari applicati a una o più istanze per generare una nuova istanza.

### 2.1 Basi dell'algebra relazionale

L'algebra relazionale essendo un'algebra è composta da un dominio e delle operazioni che danno come risultato un elemento interno al dominio.

In questo caso i domini sono le relazioni, rappresentate come un insieme di tuple e i risultati delle operazioni sono un altro insieme di tuple.

Le query sono domande sull'istanza delle relazioni.

Ci sono 3 tipi di operazioni:

- modificare una relazione: proiezione, selezione e rinomina
- operazioni su insiemi: unione, intersezione e differenza
- combinare tuple di due relazioni: prodotto cartesiano, join e theta-join

### 2.2 Modifica di una relazione

#### 2.2.1 Proiezione

##### Proiezione ( $\pi$ )

La **proiezione** esegue un taglio verticale su una relazione e sceglie un sottoinsieme degli attributi, creando una tabella con solo questi attributi.

$$\pi_{A_1, \dots, A_k}(R)$$

Prende solo le colonne di un'istanza con attributi  $A_1, \dots, A_k$ .

**Esempio:**

Customer:

Nome	C#	Città
Rossi	1	Roma
Rossi	2	Milano
Bianchi	3	Roma
Verdi	4	Roma
Rossi	5	Roma

Eseguendo  $\pi_{\text{Nome, Città}}(\text{Customer})$ :

Nome	Città
Rossi	Roma
Rossi	Milano
Bianchi	Roma
Verdi	Roma

### 2.2.2 Selezione

#### Selezione ( $\sigma$ )

La **selezione** fa un taglio orizzontale all'istanza di una relazione e sceglie tutte le tuple che rispettano una determinata condizione.

$$\sigma_C(R)$$

La condizione  $C$  è un'espressione booleana nella forma:

$$A \theta B \text{ o } A \theta a$$

In cui:

- $\theta$  è un operatore di comparazione, cioè  $\{<, >, =, \leq, \geq, \text{ecc...}\}$
- $A$  e  $B$  hanno lo stesso dominio
- $a$  è un elemento del dominio di  $A$  usato come costante

**Esempio:**

Customer:

Nome	C#	Città
Rossi	1	Roma
Rossi	2	Milano
Bianchi	3	Roma
Verdi	4	Roma
Rossi	5	Roma

Eseguendo  $\sigma_{\text{Town}=\text{Roma}}(\text{Customer})$ :

Nome	C#	Città
Rossi	1	Roma
Bianchi	3	Roma
Verdi	4	Roma
Rossi	5	Roma



### 2.2.3 Rinomina

#### Rinomina ( $\rho$ )

La **rinomina** cambia il nome di un attributo in un altro.

$$\rho_{A_1 \leftarrow A_2}(R)$$

L'attributo  $A_1$  viene rinominato in  $A_2$ .

## 2.3 Operazioni insiemistiche

Le operazioni insiemistiche possono essere fatte solo su istanze compatibili.

Due istanze sono compatibili se:

- hanno lo stesso numero di attributi
- gli attributi corrispondenti hanno lo stesso dominio

### 2.3.1 Unione

#### Unione ( $\cup$ )

L'**unione** crea una nuova istanza contenente tutte le tuple appartenenti a una delle due istanze unite.

$$r_1 \cup r_2$$

Bisogna stare attenti a non unire istanze con attributi che hanno lo stesso dominio ma che non c'entrano niente (ed esempio età e matricola).

Nel caso in cui gli attributi abbiano nomi diversi vengono presi i nomi della prima relazione.

#### Esempio:

Teachers:

Nome	Codice	Dipartimento
Rossi	1	Matematica
Bianchi	2	Fisica
Verdi	3	Inglese

Admin:

Nome	Codice	Dipartimento
Esposito	1	Matematica
Perelli	2	Informatica
Verdi	3	Inglese

Eseguendo  $AllStaff = Teachers \cup Admin$ :

Nome	Codice	Dipartimento
Rossi	1	Matematica
Bianchi	2	Fisica
Verdi	3	Inglese
Esposito	1	Matematica
Perelli	2	Informatica

In questo caso abbiamo cancellato una riga che contiene (Verdi, 3, Inglese), per sistemarlo potremmo cambiare il codice degli insegnanti in T1,T2, ecc... e quello degli admin in A1,A2, ecc...

### 2.3.2 Differenza

#### Differenza ( $-$ )

La **differenza** crea una nuova istanza contenente tutte le tuple della prima che non sono presenti nella seconda. Si identifica con  $-$ .

$$r_1 - r_2$$

La differenza al contrario di unione e intersezione non è commutativa.

#### Esempio:

Students:

Nome	Codice	Dipartimento
Rossi	1	Matematica
Bianchi	2	Inglese
Verdi	3	Informatica

Admin:

Nome	Codice	Dipartimento
Esposito	4	Informatica
Bianchi	2	Inglese
Perelli	5	Matematica

Eseguendo Students  $-$  Admins:

Nome	Codice	Dipartimento
Rossi	1	Matematica
Verdi	3	Informatica

Eseguendo Admin  $-$  Students:

Nome	Codice	Dipartimento
Esposito	4	Informatica
Perelli	5	Matematica

### 2.3.3 Intersezione

#### Intersezione ( $\cap$ )

L'**intersezione** crea una nuova istanza con le tuple in comune alle due istanze.

$$r_1 \cap r_2$$

Può anche essere definita come  $r_1 - (r_1 - r_2)$ .

## 2.4 Operazioni di combinazione

### 2.4.1 Prodotto cartesiano

#### Prodotto cartesiano ( $\times$ )

Il **prodotto cartesiano** crea una relazione con tutte le possibili combinazioni delle tuple della prima e della seconda relazione.

$$R_1 \times R_2$$

#### Esempio:

Customers:

Nome	C#	Città
Rossi	1	Roma
Rossi	2	Milano
Bianchi	3	Roma

Admin:

O#	CC#	A#	Pezzi
1	1	2	100
2	2	2	250
3	3	3	50

Eseguendo  $\text{Orders} = \text{Customer} \times \text{Order}$ :

Nome	C#	Città	O#	CC#	A#	Pezzi
Rossi	1	Roma	1	1	2	100
Rossi	1	Roma	2	2	2	250
Rossi	1	Roma	3	3	3	50
Rossi	2	Milano	1	1	2	100
Rossi	2	Milano	2	2	2	250
Rossi	2	Milano	3	3	3	50
Binchi	3	Roma	1	1	1	100
Binchi	3	Roma	2	2	2	250
Binchi	3	Roma	3	3	3	50

In questo caso ho collegato dei Customer ad alcuni Order con C# diversi sbagliando e avendo alcune tuple in eccesso, quindi eseguo una selezione solo delle tuple che hanno C# uguale a CC# e poi successivamente una proiezione per eliminare l'attributo CC# ora superfluo.

Eseguendo  $\text{Final} = \pi_{-CC\#}(\sigma_{C\#=CC\#}(\text{Customer} \times \text{Order}))$ :

Nome	C#	Città	O#	A#	Pezzi
Rossi	1	Roma	1	2	100
Rossi	2	Milano	2	2	250
Binchi	3	Roma	3	3	50

### 2.4.2 Concatenazione

#### Concatenazione ( $\bowtie$ )