

Basi di Dati in Pillole

Sofia Amarù

30 agosto 2019

Indice

1	Introduzione	4
1.1	Concetti fondamentali	4
1.1.1	Sistema informativo, sistema informatico, dati	4
1.1.2	Definizione di database, DBMS	4
1.1.3	Caratteristiche	4
1.1.4	Modelli	5
1.1.5	Transizioni, schema, istanza	5
1.1.6	Livelli di astrazione	5
1.1.7	Indipendenza dei dati	5
1.1.8	Linguaggi	5
2	Modello ER (entità-relazione)	6
2.1	Rappresentazione grafica dei costrutti	6
2.2	Costrutti	6
3	Modello Relazionale	9
3.1	Vincoli di integrità	9
3.2	Chiave, superchiave, superchiave minimale	9
4	Progettazione logica	10
4.1	Ristrutturazione schema ER	10
4.1.1	Analisi delle ridondanze	10
4.1.2	Eliminazione delle generalizzazioni	10
4.1.3	Partizionamento	11
4.1.4	Eliminazione di attributi multivalore	11
4.1.5	Accorpamento di entità	11
4.1.6	Scelta degli identificatori principali	12
4.2	Traduzione verso modello logico	12
4.2.1	Molti a molti	12
4.2.2	Uno a molti	12
4.2.3	Uno a uno con partecipazioni obbligatorie per entrambe le entità	13
4.2.4	Uno a uno con partecipazione opzionale per una sola entità	13
5	SQL	14
5.1	Interrogazioni	14
5.2	Select, from, where	14
5.3	Is null	15
5.4	Distinct, all	16
5.5	Join, alias	16
5.6	Order by	16
5.7	Operatori aggregati	17
5.8	Group by	17
5.9	Predicati su gruppi: having	17
5.10	Interrogazioni di tipo insiemistico	17
5.11	Any	18

6	Algebra relazionale	19
6.1	Operatori insiemistici	19
6.2	Operatori principali	19
6.2.1	Ridenominazione	19
6.2.2	Selezione	19
6.2.3	Proiezione	20
6.2.4	Join	20
6.2.5	Prodotto cartesiano	22
6.2.6	Theta join e equi-join	22
6.2.7	Logica a tre valori	22
6.2.8	Interrogazioni	22

Note

Il seguente testo è un personalissimo specchietto riassuntivo dei concetti fondamentali studiati durante il corso di Basi di Dati (corso di laurea triennale in Informatica presso l'Università degli Studi di Milano-Bicocca) e lo studio sullo stesso non è sufficiente per il superamento dell'esame. Si consiglia la lettura solo dopo aver studiato dal libro *Basi di Dati - Modelli e linguaggi di interrogazione* degli autori Azteni, Ceri, Paraboschi, Torlone.

Questo testo può contenere errori: nell'eventualità vi prego di contattarmi per procedere alla correzione.

Ricordo ai miei compagni che l'esame è composto da 5 esercizi relativi ai seguenti argomenti:

- Modello ER
- Modello Relazionale
- Progettazione logica
- SQL
- Algebra relazionale

1 Introduzione

1.1 Concetti fondamentali

1.1.1 Sistema informativo, sistema informatico, dati

sistema informativo: componente (non necessariamente automatizzata) di un'organizzazione, che gestisce le informazioni di interesse

sistema informatico: porzione automatizzata del sistema informativo

- acquisizione e memorizzazione
- aggiornamento
- interrogazione
- elaborazione

dato \neq informazione

- dato: non ha alcun valore fino a quando non viene interpretato (immutato nel tempo)
- informazione: si ha quando un dato viene interpretato

1.1.2 Definizione di database, DBMS

DATABASE: collezione di dati usati per rappresentare informazioni di interesse

DBMS (data base management system): software per la gestione di DB

1.1.3 Caratteristiche

caratteristiche di un DB:

- privacy
- affidabilità
- efficienza
- efficacia

fasi:

- definizione
- creazione e popolazione
- manipolazione

caratteristiche di un sistema di DB:

- natura autodescrittiva
- separazione tra programmi e dati
- astrazione dei dati (visione concettuale del DB per l'utente)
- supporto viste multiple dei dati
- condivisione dei dati e gestione con utenti multipli

1.1.4 Modelli

modello logico:

- relazionale (costruttore relazione; record a struttura fissa; tabella)
- gerarchico (strutture ad albero)
- reticolare (grafi)
- a oggetti
- XML

modello concettuale:

- ER (entità-relazione)

1.1.5 Transizioni, schema, istanza

transizione: insieme indivisibile di operazioni

schema: struttura, invariante nel tempo

istanza: valori attuali, possono cambiare anche molto rapidamente

1.1.6 Livelli di astrazione

schema logico - descrizione dell'intera base di dati per mezzo del modello logico

schema interno - rappresentazione dello schema logico per mezzo di strutture fisiche di memorizzazione

schema esterno - descrizione di una porzione della base di dati per mezzo del modello logico. A volte non è esplicitamente presente ma è possibile definire relazioni derivate (viste)

1.1.7 Indipendenza dei dati

indipendenza fisica - interagire col DBMS indipendentemente dalla struttura fisica dei dati

indipendenza logica - interagire col livello esterno indipendentemente dal livello logico




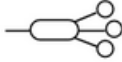


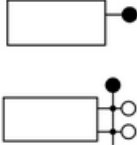



1.1.8 Linguaggi

DDL Data Definition Language - definizione degli schemi e delle autorizzazioni per l'accesso

DML Data Manipulation Language - interrogazione e aggiornamento delle istanze del DB

2 Modello ER (entità-relazione)

2.1 Rappresentazione grafica dei costrutti

Costrutto	Rappresentazione grafica
entità	
relazione	
attributo semplice	
attributo composto	
cardinalità di relazione	
cardinalità di attributo	
identificatore interno	
identificatore esterno	
generalizzazione	
sottoinsieme	

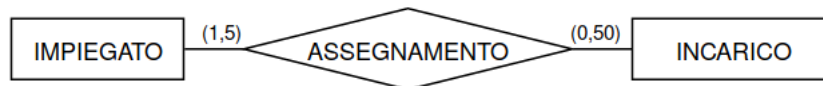
2.2 Costrutti

entità - classe di oggetti con proprietà comuni (il nome va al singolare)

relazione - legame logico (il nome deve essere univoco; è preferibile usare un sostantivo per evitare di assegnare un verso)

attributo composto - raggruppamento di attributi (Es. Via, Numero, CAP)

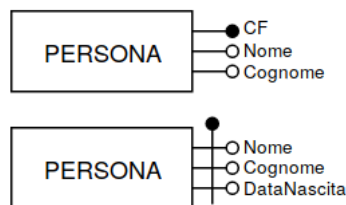
cardinalità di relazione - indica quante volte, in una relazione, l'occorrenza dell'entità è legata ad occorrenze dell'altra entità



un impiegato ha da 1 a 5 incarichi

identificatore (chiave) - identificatore univoco dell'entità

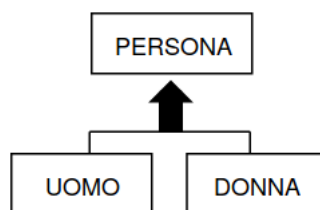
identificatore interno - se è uno o più attributi di un'entità



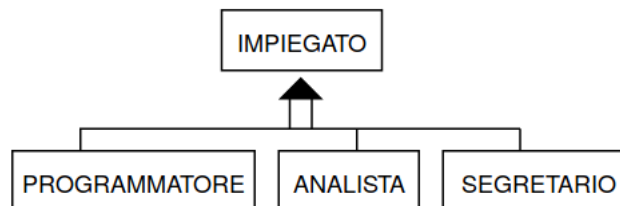
identificatore esterno - se un'entità viene identificata da un attributo di un'altra entità con cui ha una relazione



generalizzazione totale - non esistono altri figli



generalizzazione parziale - ci possono essere altri figli



sottoinsieme - un solo figlio

generalizzazione esclusiva - es. uomo-donna (o è uomo o è donna)

generalizzazione sovrapposta - es. studente-lavoratore (può essere studente, lavoratore o studente-lavoratore)

3 Modello Relazionale

3.1 Vincoli di integrità

intrarelazionale - soddisfatto rispetto a singole relazioni

di tupla - può essere valutato su ciascuna tupla indipendentemente dalle altre.

Es. la lode può comparire solo con voto 30

di dominio - restrizione sul dominio dell'attributo

Es. voto compreso tra 18 e 30

interrelazionale - coinvolge più relazioni

vincolo di integrità referenziale - proprietà dei dati che per essere soddisfatta richiede che ogni valore di un attributo (colonna) di una relazione (tabella) esista come valore di un altro attributo in un'altra relazione.

Es. matricola compare in ESAMI solo se compare in STUDENTI

3.2 Chiave, superchiave, superchiave minimale

chiave - insieme di attributi utilizzato per identificare univocamente le tuple di una relazione

superchiave - insieme di attributi di una relazione tali che le relative tuple sono tutte diverse tra loro (identificazione univoca)

superchiave minimale - superchiave che non contiene una superchiave

chiave primaria - chiave a cui non è permesso contenere valori nulli (la presenza di NULL impedisce l'identificazione univoca)

4 Progettazione logica

Fasi

- **Ristrutturazione schema ER**
 - Analisi delle ridondanze
 - Eliminazione delle generalizzazioni
 - Partizionamento/accorpamento entità e associazioni
 - Scelta identificatori principali
- **Traduzione verso modello logico**

4.1 Ristrutturazione schema ER

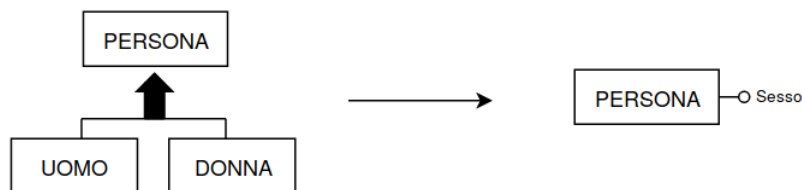
4.1.1 Analisi delle ridondanze

La presenza di un dato ridondante comporta:

- × occupazione memoria
- × operazioni per mantenere dato aggiornato
- ✓ riduzione accessi necessari per calcolarlo

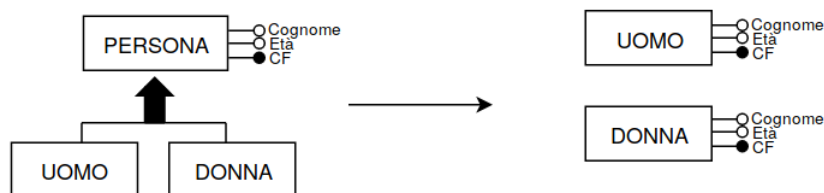
4.1.2 Eliminazione delle generalizzazioni

Accorpamento dei figli nel genitore



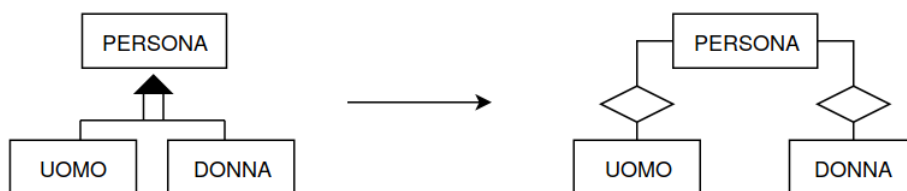
Accorpamento del genitore dei figli

NB. Possibile solo se la generalizzazione è totale!



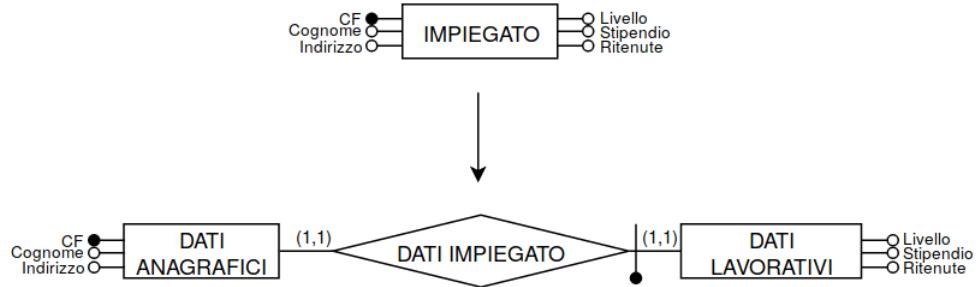
Sostituzione generalizzazioni con associazioni

NB. Conveniente con generalizzazione parziale.

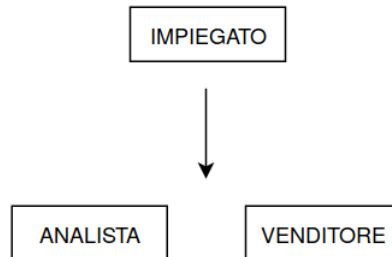


4.1.3 Partizionamento

decomposizione verticale

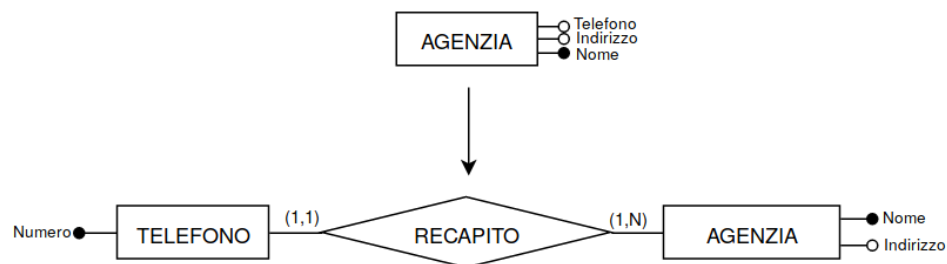


decomposizione orizzontale



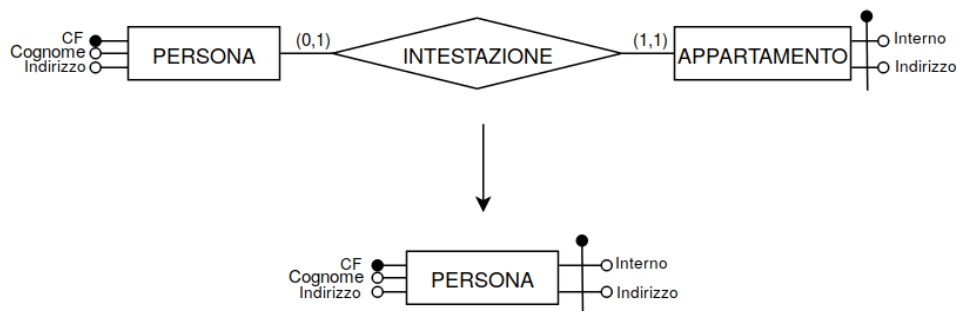
4.1.4 Eliminazione di attributi multivalore

Es. Un'agenzia può avere più numeri di telefono



4.1.5 Accorpamento di entità

NB. Solitamente su associazioni uno a uno

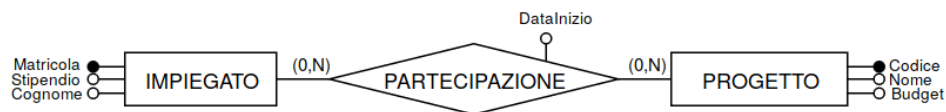


4.1.6 Scelta degli identificatori principali

- No attributi con valori nulli
- Preferire identificatore composto da 1 o **pochi** attributi
- Preferire identificatore interno ad identificatore esterno
- Preferire identificatore che viene utilizzato da molte operazioni per accedere alle occorrenze dell'entità

4.2 Traduzione verso modello logico

4.2.1 Molti a molti



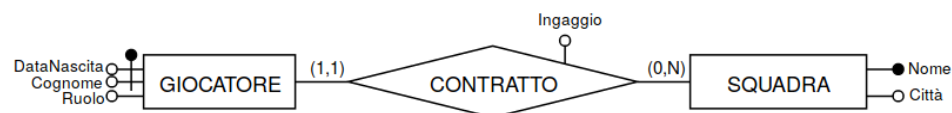
IMPIEGATO(Matricola, Cognome, Stipendio)

PROGETTO(Codice, Nome, Budget)

PARTECIPAZIONE(Impiegato, Progetto, DataInizio)

NB. Impiegato è la ridenominazione di Matricola e Progetto è la ridenominazione di Codice

4.2.2 Uno a molti



GIOCATORE(DataNascita, Cognome, Ruolo)

SQUADRA(Nome, Città)

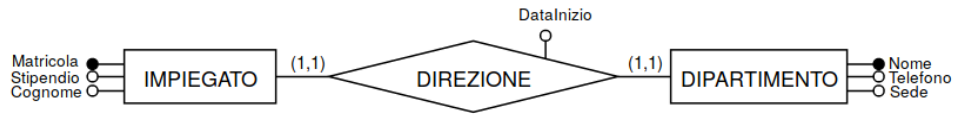
CONTRATTO(Giocatore, Squadra, Ingaggio)

diventa

GIOCATORE(DataNascita, Cognome, Ruolo, Squadra, Ingaggio)

SQUADRA(Nome, Città)

4.2.3 Uno a uno con partecipazioni obbligatorie per entrambe le entità



DIRETTORE(Matricola, Cognome, Stipendio, Dipartimento, InizioDirezione)

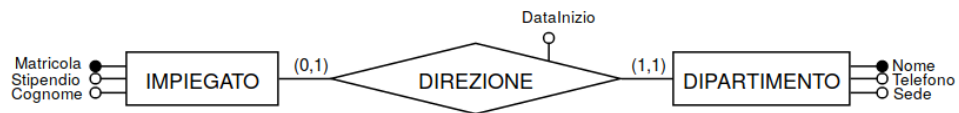
DIPARTIMENTO(Nome, Telefono, Sede)

oppure

DIRETTORE(Matricola, Cognome, Stipendio)

DIPARTIMENTO(Nome, Telefono, Sede, Direttore, InizioDirezione)

4.2.4 Uno a uno con partecipazione opzionale per una sola entità



IMPIEGATO(Matricola, Cognome, Stipendio)

DIPARTIMENTO(Nome, Telefono, Sede)

DIREZIONE(Direttore, Dipartimento, DataInizio)

5 SQL

Structured Query Language, contiene funzionalità di un **DDL** (Data Definition Language) e di un **DML** (Data Manipulation Language)

5.1 Interrogazioni

```
select ListaAttributi  
from ListaTabelle  
[ where Condizione ]
```

esempio di interrogazione

5.2 Select, from, where

clausola **select** (target list) - specifica gli elementi dello schema della tabella risultato. Come argomento può comparire *, che rappresenta tutti gli attributi delle tabelle elencate nella clausola from.

clausola **from** - ha come argomento l'insieme delle tabelle a cui si vuole accedere. Sul prodotto cartesiano delle tabelle vengono applicate le condizioni contenute nella clausola where.

clausola **where** - ha come argomento un'espressione booleana

Consideriamo le due tabelle

IMPIEGATO(Nome, Cognome, Dipart, Ufficio, Stipendio, Città)

DIPARTIMENTO(Nome, Indirizzo, Città)

1. estrarre lo stipendio degli impiegati di nome "Rossi".

```
select Stipendio as Salario  
from Impiegato  
where Cognome = 'Rossi'
```

2. estrarre tutte le informazioni relative agli impiegati di cognome "Rossi"

```
select *  
from Impiegato  
where Cognome = 'Rossi'
```

3. estrarre lo stipendio mensile dell'impiegato che ha cognome "Bianchi"

```
select Stipendio/12 as StipendioMensile  
from Impiegato  
where Cognome = 'Bianchi'
```

4. estrarre i nomi degli impiegati e le città in cui lavorano

```
select Impiegato.Nome, Impiegato.Cognome, Dipartimento.Città
from Impiegato, Dipartimento
where Impiegato.Dipart = Dipartimento.Nome
```

5. gli attributi per cui sorge un'ambiguità sono Nome e Città. L'interrogazione precedente può essere espressa facendo uso degli alias per le tabelle allo scopo di abbreviare i riferimenti a esse.

```
select I.Nome, Cognome, D.Città
from Impiegato as I, Dipartimento as D
where Dipart = D.Nome
```

6. estrarre nome e cognome degli impiegato che lavorano nell'ufficio 20 del dipartimento Amministrazione.

```
select Nome, Cognome
from Impiegato
where Ufficio = 20 and Dipart = 'Amministrazione'
```

7. estrarre i nomi e i cognomi degli impiegati che lavorano nel dipartimento Amministrazione o nel dipartimento Produzione

```
select Nome, Cognome
from Impiegato
where Dipart = 'Amministrazione' or
      Dipart = 'Produzione'
```

8. estrarre i nomi propri degli impiegati di nome "Rossi" che lavorano nei dipartimenti Amministrazione o Produzione

```
select Nome
from Impiegato
where Cognome = 'Rossi' and
      (Dipart = 'Amministrazione' or
       Dipart = 'Produzione')
```

5.3 Is null

predicato `is null` - per selezionare termini con valori nulli (\neg `is not null`)

5.4 Distinct, all

parola chiave `distinct` - per eliminare i duplicati (\neg `all`, di default, opzionale)

Consideriamo la relazione

PERSONA(CodFiscale, Nome, Cognome, Città)

9. estrarre le città delle persone con cognome “Rossi”, facendo comparire ogni città al più una volta

```
select distinct Città
from Persona
where Cognome = 'Rossi'
```

5.5 Join, alias

operatore `join` - si scrive nell’ambito della clausola from, e non compare il where. (inner, right, left, full)

10. riscrivi il punto 5.

```
select I.Nome, Cognome, D.Città
from Impiegato I join Dipartimento D
on Dipart = D.Nome
```

11. estrarre tutti gli impiegati che hanno lo stesso cognome (ma diverso nome) di impiegati del dipartimento Produzione

```
select I1.Nome, I1.Cognome
from Impiegato I1, Impiegato I2
where I1.Cognome = I2.Cognome and
      I1.Nome <> I2.Nome and
      I2.Dipart = 'Produzione'
```

immaginiamo che al momento della definizione degli `alias` (I1 e I2) vengano create due diverse tabelle che verranno confrontate tra di loro.

5.6 Order by

clausola `order by` - ordina in modo ascendente (`asc`) o discendente (`desc`)

12. estrarre il contenuto della tabella AUTOMOBILE ordinato in base alla marca (in modo discendente) e al modello

```
select *
from Automobile
order by Marca desc, Modello
```

5.7 Operatori aggregati

`count` - torna il numero degli attributi
`sum` - torna la somma dei valori dell'espressione
`max` - massimo
`min` - minimo
`avg` - media dei valori

13. estrarre il numero di diversi valori dell'attributo Stipendio fra tutte le righe di IMPIEGATO

```
select count (distinct Stipendio)
from Impiegato
```

5.8 Group by

clausola `group by` - raggruppa per

14. estrarre la somma degli stipendi di tutti gli impiegati dello stesso dipartimento

```
select Dipart, sum(Stipendio)
from Impiegato
group by Dipartimento
```

5.9 Predicati su gruppi: having

predicato `having` - che ha

15. estrarre i dipartimenti che spendono più di 100 mila euro in stipendi

```
select Dipart, sum(Stipendio) as SommaStipendi
from Impiegato
group by Dipart
having sum(Stipendio) > 100
```

5.10 Interrogazioni di tipo insiemistico

`union`
`intersect`
`except`

16. estrarre i nomi e i cognomi degli impiegati

```
select Nome
from Impiegato
union
select Cognome
from Impiegato
```

5.11 Any

clausola **any** - la riga soddisfa la condizione se è vero il confronto tra il valore dell'attributo per la riga e almeno uno degli elementi restituiti dall'interrogazione

17. estrarre gli impiegati che lavorano in dipartimenti situati a Firenze

```
select *  
from Impiegato  
where Dipart = any (select Nome  
                    from Dipartimento  
                    where Città = 'Firenze')
```

6 Algebra relazionale

Relazione: insieme di tuple omogenee.

6.1 Operatori insiemistici

unione \cup

intersezione \cap

differenza $-$

LAUREATI

Matricola	Cognome	Età
7274	Rossi	37
7432	Neri	39
9824	Verdi	38

DIRIGENTI

Matricola	Cognome	Età
9297	Neri	56
7432	Neri	39
9824	Verdi	38

LAUREATI \cup DIRIGENTI

Matricola	Cognome	Età
7274	Rossi	37
7432	Neri	39
9824	Verdi	38
9297	Neri	56

LAUREATI \cap DIRIGENTI

Matricola	Cognome	Età
7432	Neri	39
9824	Verdi	38

LAUREATI $-$ DIRIGENTI

Matricola	Cognome	Età
7274	Rossi	37

6.2 Operatori principali

6.2.1 Ridenominazione

Agisce solo sullo schema, cambiando solo i nomi degli attributi

$$\rho_{\text{nuovo nome} \leftarrow \text{vecchio nome}}$$

PATERNITÀ

Padre	Figlio
Adamo	Caino
Adamo	Abele
Abramo	Isacco
Isacco	Giacobbe

$\rho_{\text{Genitore} \leftarrow \text{Padre}}(\text{PATERNITÀ})$

Genitore	Figlio
Adamo	Caino
Adamo	Abele
Abramo	Isacco
Isacco	Giacobbe

6.2.2 Selezione

Produce un sottoinsieme delle tuple su tutti gli attributi (decomposizione orizzontale)

$$\sigma_{\text{condizione di selezione}}$$

IMPIEGATI

Cognome	Nome	Età	Stipendio
Rossi	Mario	25	2.000,00
Neri	Luca	40	3.000,00
Verdi	Nico	36	4.500,00
Rossi	Marco	40	3.900,00

 $\sigma_{\text{Età} > 30 \wedge \text{Stipendio} > 4.000,00}(\text{IMPIEGATI})$

Cognome	Nome	Età	Stipendio
Verdi	Nico	36	4.500,00

6.2.3 Proiezione

È l'insieme delle tuple ottenute considerando solo i valori dei campi scelti. La proiezione contiene lo stesso numero di tuple della relazione di partenza solo se i campi scelti sono una superchiave. (decomposizione verticale)

$\pi_{\text{campi scelti}}$

IMPIEGATI

Cognome	Nome	Reparto	Capo
Rossi	Mario	Vendite	Gatti
Neri	Luca	Vendite	Gatti
Verdi	Mario	Personale	Lupi
Rossi	Marco	Personale	Lupi

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

Cognome	Nome
Rossi	Mario
Neri	Luca
Verdi	Mario
Rossi	Marco

Figura 3.9 Una proiezione

IMPIEGATI

Cognome	Nome	Reparto	Capo
Rossi	Mario	Vendite	Gatti
Neri	Luca	Vendite	Gatti
Verdi	Mario	Personale	Lupi
Rossi	Marco	Personale	Lupi

 $\pi_{\text{Reparto, Capo}}(\text{IMPIEGATI})$

Reparto	Capo
Vendite	Gatti
Personale	Lupi

Figura 3.10 Una proiezione con meno tuple dell'operando

6.2.4 Join

Permette di correlare dati contenuti in relazioni diverse

⋈

join naturale - correla dati in relazioni diverse sulla base di valori uguali in campi uguali.

r_1	Impiegato	Reparto
	Rossi	vendite
	Neri	produzione
	Bianchi	produzione

r_2	Reparto	Capo
	produzione	Mori
	vendite	Bruni

$r_1 \bowtie r_2$	Impiegato	Reparto	Capo
	Rossi	vendite	Bruni
	Neri	produzione	Mori
	Bianchi	produzione	Mori

join completo - ogni tupla contribuisce ad almeno una tupla del risultato (vedi immagine sopra)

join con tuple dangling - alcune tuple non contribuiscono al risultato

r_1	Impiegato	Reparto
	Rossi	vendite
	Neri	produzione
	Bianchi	produzione

r_2	Reparto	Capo
	produzione	Mori
	acquisti	Bruni

$r_1 \bowtie r_2$	Impiegato	Reparto	Capo
	Neri	produzione	Mori
	Bianchi	produzione	Mori

join esterni - tutte le tuple contribuiscono al risultato, eventualmente estese con valori nulli

- sinistro
- destro
- completo

r_1	Impiegato	Reparto
	Rossi	vendite
	Neri	produzione
	Bianchi	produzione

r_2	Reparto	Capo
	produzione	Mori
	acquisti	Bruni

$r_1 \bowtie_{\text{LEFT}} r_2$	Impiegato	Reparto	Capo
	Rossi	vendite	NULL
	Neri	produzione	Mori
	Bianchi	produzione	Mori

$r_1 \bowtie_{\text{RIGHT}} r_2$	Impiegato	Reparto	Capo
	Neri	produzione	Mori
	Bianchi	produzione	Mori
	NULL	acquisti	Bruni

$r_1 \bowtie_{\text{FULL}} r_2$	Impiegato	Reparto	Capo
	Rossi	vendite	NULL
	Neri	produzione	Mori
	Bianchi	produzione	Mori
	NULL	acquisti	Bruni

6.2.5 Prodotto cartesiano

Il risultato è la combinazione delle tuple in tutti i modi possibili.

NB. il termine è improprio: il prodotto cartesiano di due insiemi è un insieme di coppie, qui abbiamo tuple.

IMPIEGATI		PROGETTI	
Impiegato	Progetto	Codice	Nome
Rossi	A	A	Venere
Neri	A	B	Marte
Neri	B		

IMPIEGATI \bowtie PROGETTI			
Impiegato	Progetto	Codice	Nome
Rossi	A	A	Venere
Neri	A	A	Venere
Neri	B	A	Venere
Rossi	A	B	Marte
Neri	A	B	Marte
Neri	B	B	Marte

6.2.6 Theta join e equi-join

theta join - operatore derivato, prodotto cartesiano + selezione

equi-join - theta join in cui la condizione di selezione è una congiunzione di atomi di uguaglianza con un attributo della prima relazione

IMPIEGATI		PROGETTI	
Impiegato	Progetto	Codice	Nome
Rossi	A	A	Venere
Neri	A	B	Marte
Neri	B		

$\sigma_{\text{Progetto}=\text{Codice}}(\text{IMPIEGATI} \bowtie \text{PROGETTI})$			
Impiegato	Progetto	Codice	Nome
Rossi	A	A	Venere
Neri	A	A	Venere
Neri	B	B	Marte

6.2.7 Logica a tre valori

vero (V), falso (F), sconosciuto (U)

<i>not</i>		<i>and</i>	V	U	F	<i>or</i>	V	U	F
F	V	V	V	U	F	V	V	V	V
U	U	U	U	U	F	U	V	U	U
V	F	F	F	F	F	F	V	U	F

6.2.8 Interrogazioni

Interrogazione: funzione che, applicata alla base di dati, produce relazioni.