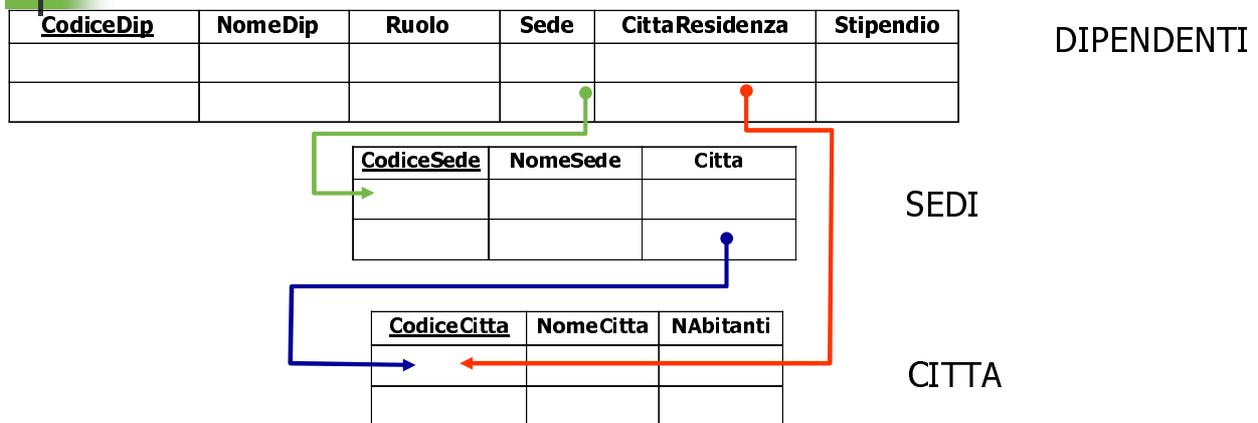


I dipendenti

- Progettare lo schema relazionale che memorizzi lo i dipendenti di un azienda, memorizzando:
 - per ogni dipendente, nome, ruolo, stipendio, sede e città di residenza;
 - per ogni sede, il nome della sede e la città dove si trova
 - per ogni città, il nome e il numero di abitanti

1

I dipendenti

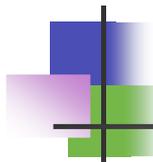


DIPENDENTI(CodiceDip, NomeDip, Ruolo, Sede, CittaResidenza, stipendio)

SEDI(CodiceSede, NomeSede, Citta)

CITTA(CodiceCitta, NomeCitta, NAbitanti)

2

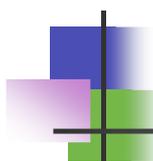


I dipendenti

Scrivere in algebra relazionale con join naturale, algebra relazionale con theta-join, calcolo sui domini, calcolo sulle tuple, le seguenti interrogazioni:

1. cercare le città in cui lavorano e il nome di tutti i dipendenti che guadagnano più di 1700
2. cercare il nome dei dipendenti che non sono residenti a Siena
3. cercare il nome delle sedi che non hanno dipendenti residenti a Siena
4. cercare il nome delle sedi che hanno almeno due dipendenti
5. cercare il nome delle sedi che hanno esattamente un dipendente

3



Interrogazione 1: algebra relazionale

cercare le città in cui lavorano e il nome di tutti i dipendenti che guadagnano più di 1700

Algebra relazionale con join naturale

$$\pi_{\text{NomeDip, NomeCitta}}(\sigma_{\text{Stipendio} > 1700}(\text{DIPENDENTI}) \bowtie \rho_{\text{Sede} \leftarrow \text{CodiceSede}}(\text{SEDI}) \bowtie \rho_{\text{Citta} \leftarrow \text{CodiceCitta}}(\text{CITTA}))$$

Algebra relazionale con theta-join

$$\pi_{\text{NomeDip, NomeCitta}}(\sigma_{\text{Stipendio} > 1700}(\text{DIPENDENTI}) \bowtie_{\text{Sede} = \text{CodiceSede}} \text{SEDI} \bowtie_{\text{Citta} = \text{CodiceCitta}} (\text{CITTA}))$$

4

Interrogazione 1: calcolo su domini

cercare le città in cui lavorano e il nome di tutti i dipendenti che guadagnano più di 1700

{Citta: c, Nome: n | DIPENDENTI(CodiceDip: cod, NomeDip: n, Ruolo: r, Sede: s, CittaResidenza: cr, Stipendio: st)
^ (st > 1700)
^ SEDI(CodiceSede: s, NomeSede: ns, Citta: cc)
^ CITTA(CodiceCitta: cc, NomeCitta: c, NAbitanti: na)}

5

Interrogazione 1: calcolo su tuple

cercare le città in cui lavorano e il nome di tutti i dipendenti che guadagnano più di 1700

{Citta: c.(NomeCitta), Nome: d.(NomeDip) | d(DIPENDENTI), s(SEDI), c(CITTA) |
d.Sede = s.CodiceSede ^
s.Citta = c.CodiceCitta ^
d.Stipendio > 1700 }

6

Interrogazione 2: algebra relazionale

cercare il nome dei dipendenti che non sono residenti a siena

Algebra relazionale con join naturale

$DIPENDENTI_DI_SIENA = \pi_{NomeDip} (DIPENDENTI \bowtie \rho_{CittaResidenza \leftarrow CodiceCitta} (\sigma_{NomeCitta='Siena'}(CITTA)))$
interrogazione $\pi_{NomeDip} (DIPENDENTI) - DIPENDENTI_DI_SIENA$

Algebra relazionale con theta-join

$DIPENDENTI_DI_SIENA = \pi_{NomeDip} (DIPENDENTI \bowtie_{CittaResidenza=CodiceCitta} (\sigma_{NomeCitta='Siena'}(CITTA)))$
interrogazione $\pi_{NomeDip} (DIPENDENTI) - DIPENDENTI_DI_SIENA$

7

Interrogazione 2: calcolo su domini

cercare il nome dei dipendenti che non sono residenti a siena

$\{Nome: n \mid DIPENDENTI(CodiceDip: cod, NomeDip: n, Ruolo: r, Sede: s, CittaResidenza: cr, Stipendio: st) \wedge CITTA(CodiceCitta: cr, NomeCitta: c, NAbitanti: na) \wedge c \neq 'Siena'\}$

8

Interrogazione 2: calcolo su tuple

cercare il nome dei dipendenti che non sono residenti a siena

$$\{ \text{Nome: d.}(\text{NomeDip}) \mid \text{d}(\text{DIPENDENTI}), \text{c}(\text{CITTA}) \mid \\ \text{d.CittaResidenza}=\text{c.CodiceCitta} \wedge \\ \text{c.NomeCitta} \neq \text{'Siena'} \}$$

9

Interrogazione 3: algebra relazionale

cercare il nome delle sedi che non hanno dipendenti residenti a Siena

Algebra relazionale con join naturale

$$\text{SEDI_CON_DIP_SIENA} = \pi_{\text{CodiceSede, NomeSede}} (\text{SEDI} \bowtie_{\rho_{\text{CodiceSede} \leftarrow \text{Sede}}} (\text{DIPENDENTI} \bowtie_{\rho_{\text{CittaResidenza} \leftarrow \text{CodiceCitta}}} (\sigma_{\text{NomeCitta} = \text{'Siena'}}(\text{CITTA}))))$$

Interrogazione: $\pi_{\text{NomeSede}} (\pi_{\text{CodiceSede, NomeSede}}(\text{SEDI}) - \text{SEDI_CON_DIP_SIENA})$

Algebra relazionale con theta-join

$$\text{SEDI_CON_DIP_SIENA} = \pi_{\text{CodiceSede, NomeSede}} (\text{SEDI} \bowtie_{\text{CodiceSede} = \text{Sede}} (\text{DIPENDENTI} \bowtie_{\text{CittaResidenza} = \text{CodiceCitta}} (\sigma_{\text{NomeCitta} = \text{'Siena'}}(\text{CITTA}))))$$

Interrogazione: $\pi_{\text{NomeSede}} (\pi_{\text{CodiceSede, NomeSede}}(\text{SEDI}) - \text{SEDI_CON_DIP_SIENA})$

10

Interrogazione 3: calcolo su domini

cercare il nome delle sedi che non hanno dipendenti residenti a Siena

$$\{ \text{Nome: ns} \mid \text{SEDI}(\text{CodiceSede: s, NomeSede: ns, Citta: cc}) \\ \wedge \neg (\exists \text{cod}(\exists n(\exists r(\exists cr(\exists st(\exists c(\exists na \\ \text{DIPENDENTI}(\text{CodiceDip: cod, NomeDip: n, Ruolo: r, Sede: s, \\ \text{CittaResidenza: cr, Stipendio: st}) \\ \wedge \text{CITTA}(\text{CodiceCitta: cr, NomeCitta: c, NAbitanti: na}) \\ \wedge c = \text{'Siena'}))))))))) \}$$

11

Interrogazione 3: calcolo su tuple

cercare il nome delle sedi che non hanno dipendenti residenti a Siena

$$\{ \text{Nome: s.}(\text{NomeSede}) \mid s(\text{SEDI}) \mid \\ \neg \exists c(\text{CITTA}) (\exists d(\text{DIPENDENTI}) \\ d.\text{Sede} = s.\text{CodiceSede} \wedge \\ d.\text{CittaResidenza} = c.\text{CodiceCitta} \wedge \\ c.\text{NomeCitta} = \text{'Siena'}) \}$$

12

Interrogazione 4: algebra relazionale

cercare il nome delle sedi che hanno almeno due dipendenti

Algebra relazionale con join naturale

$DIPENDENTI1 = \rho_{\text{CodiceDip1} \leftarrow \text{CodiceDip}, \text{NomeDip1} \leftarrow \text{NomeDip}, \text{Ruolo1} \leftarrow \text{Ruolo}, \text{CittaResidenza1} \leftarrow \text{CittaResidenza}, \text{Stipendio1} \leftarrow \text{Stipendio}}(DIPENDENTI)$

$DIPENDENTI2 = \rho_{\text{CodiceDip2} \leftarrow \text{CodiceDip}, \text{NomeDip2} \leftarrow \text{NomeDip}, \text{Ruolo2} \leftarrow \text{Ruolo}, \text{CittaResidenza2} \leftarrow \text{CittaResidenza}, \text{Stipendio2} \leftarrow \text{Stipendio}}(DIPENDENTI)$

Interrogazione: $\pi_{\text{NomeSede}}(SEDI \bowtie \rho_{\text{CodiceSede} \leftarrow \text{Sede}}(\sigma_{\text{CodiceDip1} \neq \text{CodiceDip2}}(DIPENDENTI1 \bowtie DIPENDENTI2)))$

13

Interrogazione 4: algebra relazionale

cercare il nome delle sedi che hanno almeno due dipendenti

Algebra relazionale con theta-join

$DIPENDENTI1 = \rho_{\text{CodiceDip1} \leftarrow \text{CodiceDip}, \text{NomeDip1} \leftarrow \text{NomeDip}, \text{Ruolo1} \leftarrow \text{Ruolo}, \text{CittaResidenza1} \leftarrow \text{CittaResidenza}, \text{Stipendio1} \leftarrow \text{Stipendio}}(DIPENDENTI)$

$DIPENDENTI2 = \rho_{\text{CodiceDip2} \leftarrow \text{CodiceDip}, \text{NomeDip2} \leftarrow \text{NomeDip}, \text{Ruolo2} \leftarrow \text{Ruolo}, \text{CittaResidenza2} \leftarrow \text{CittaResidenza}, \text{Stipendio2} \leftarrow \text{Stipendio}}(DIPENDENTI)$

Interrogazione: $\pi_{\text{NomeSede}}(SEDI \bowtie_{\text{CodiceDip1} \neq \text{CodiceDip2}} \rho_{\text{CodiceSede} \leftarrow \text{Sede}}(DIPENDENTI1 \bowtie_{\text{CodiceDip1} \neq \text{CodiceDip2}} DIPENDENTI2))$

14

Interrogazione 4: calcolo su domini

cercare il nome delle sedi che hanno almeno due dipendenti

```
{Nome: ns | SEDI(CodiceSede: s, NomeSede: ns, Citta: cc) ^  
  DIPENDENTI(CodiceDip: cod1, NomeDip: n1, Ruolo: r1, Sede:s,  
              CittaResidenza:cr1, Stipendio: st1) ^  
  DIPENDENTI(CodiceDip: cod2, NomeDip: n2, Ruolo: r2, Sede:s,  
              CittaResidenza:cr2, Stipendio: st2) ^  
  ¬ (cod1=cod2)  
}
```

15

Interrogazione 3: calcolo su tuple

cercare il nome delle sedi che hanno almeno due dipendenti

```
{Nome: s.(NomeSede) | s(SEDI),d1(DIPENDENTI),d2(DIPENDENTI)) |  
  d1.Sede=s.CodiceSede ^  
  d2.Sede=s.CodiceSede ^  
  ¬ (d1.cod=d2.cod)  
}
```

16

Interrogazione 5: algebra relazionale

cercare il nome delle sedi che hanno esattamente un dipendente

Algebra relazionale con join naturale

$SEDI_SENZA_DIPENDENTI = \pi_{\text{CodiceSede}}(SEDI) - \rho_{\text{CodiceSede} \leftarrow \text{Sede}}(\pi_{\text{Sede}}(DIPENDENTI))$

$SEDI_CON_ALMENO_DUE_DIP = \dots$ vedi esercizio precedente

Interrogazione: $\pi_{\text{NomeSede}}(SEDI \bowtie ((SEDI - SEDI_SENZA_DIPENDENTI) - SEDI_CON_ALMENO_DUE_DIP))$

Algebra relazionale con theta-join

Interrogazione: $\pi_{\text{NomeSede}}(SEDI \bowtie_{\text{Sede}=\text{Sede}} ((SEDI - SEDI_SENZA_DIPENDENTI) - SEDI_CON_ALMENO_DUE_DIP))$

17

Interrogazione 5: calcolo su domini

cercare il nome delle sedi che hanno esattamente un dipendente

$\{ \text{Nome: ns} \mid SEDI(\text{CodiceSede: } s, \text{NomeSede: } ns, \text{Citta: } cc) \wedge$

$DIPENDENTI(\text{CodiceDip: } cod1, \text{NomeDip: } n1, \text{Ruolo: } r1, \text{Sede: } s,$
 $\text{CittaResidenza: } cr1, \text{Stipendio: } st1) \wedge$

$\neg (\exists cod2 (\exists n2 (\exists r2 (\exists cr2 (\exists st2$

$DIPENDENTI(\text{CodiceDip: } cod2, \text{NomeDip: } n2, \text{Ruolo: } r2, \text{Sede: } s,$
 $\text{CittaResidenza: } cr2, \text{Stipendio: } st2) \wedge$

$\neg (cod1 = cod2))))))$

$\}$

18

Interrogazione 5: calcolo su tuple

cercare il nome delle sedi che hanno esattamente un dipendente

```
{Nome: s.(NomeSede) | s(SEDI),d1(DIPENDENTI),d2(DIPENDENTI) |  
  d1.Sede=s.CodiceSede ∧  
  ¬ (∃d2(DIPENDENTI)  
    d2.Sede=s.CodiceSede ∧  
    ¬ (d1.cod=d2.cod))  
}
```