

## Soluzione Esercitazione 2

### Esercizio 1

Estrarre gli studenti che hanno sostenuto almeno due esami di corsi diversi.

*Calcolo Relazionale*

$$\{t | \exists t_1 \in ESAME, \exists t_2 \in ESAME (t_1[matr] = t_2[matr] \wedge t_1[cod\_corso] \neq t_2[cod\_corso] \wedge t[matr] = t_1[matr])\}$$

*Datalog*

$$ALMENODUE(m) : -ESAME(m, cc1, -, -), ESAME(m, cc2, -, -), cc1 \neq cc2 \\ ? - ALMENODUE(x)$$

### Esercizio 2

Estrarre gli studenti che non hanno superato nessun esame nel 2017.

*Calcolo Relazionale*

$$\{t | \exists t_1 \in STUDENTE (t[matr] = t_1[matr] \wedge \neg (\exists t_2 \in ESAME \\ (t_2[data] \geq 01/01/2017 \wedge t_2[data] \leq 31/12/2017 \wedge t_2[voto] \geq 18 \wedge t_1[matr] = t_2[matr])))\}$$

*Datalog*

$$ALMENOUNO17(m) : -ESAME(m, -, d, v), d \geq 01/01/2017, d \leq 31/12/2017, v \geq 18 \\ NOESAME17(m) : -STUDENTE(m, -, -, -), \neg ALMENOUNO17(m) \\ ? - NOESAME17(x)$$

### Esercizio 3

Per ogni corso, estrarre gli studenti che hanno superato un esame nell'ultima seduta svolta.

*Calcolo Relazionale*

$$\{t | \exists t_1 \in ESAME (t[matr, cod\_corso] = t_1[matr, cod\_corso] \wedge \neg (\exists t_2 \in ESAME, \\ (t_1[cod\_corso] = t_2[cod\_corso] \wedge t_1[data] < t_2[data])) \wedge t_1[voto] \geq 18)\}$$

*Datalog*

$$APPELLI\_PREC(c, d_1) : -ESAME(c, -, d_1), ESAME(-, c, -, d_2), d_2 > d_1 \\ ULTIMO\_APPELLO(m, c) : -ESAME(m, c, d, v), \neg APPELLI\_PREC(c, d), v \geq 18 \\ ? - ULTIMO\_APPELLO(x, y)$$

### Esercizio 4

Estrarre gli studenti che hanno sostenuto almeno tre esami di corsi diversi.

### Calcolo Relazionale

$$\{t|\exists t_1 \in ESAME(t[matr] = t_1[matr] \wedge (\exists t_2 \in ESAME(t_1[matr] = t_2[matr] \wedge t_1[cod\_corso] \neq t_2[cod\_corso] \wedge (\exists t_3 \in ESAME(t_2[matr] = t_3[matr] \wedge t_3[cod\_corso] \neq t_2[cod\_corso] \wedge t_3[cod\_corso] \neq t_1[cod\_corso])))))\}$$

### Datalog

$$ALMENO\_TRE(m) : -ESAME(m, c_1, -, -), ESAME(m, c_2, -, -), ESAME(m, c_3, -, -), \\ c_1 \neq c_2, c_1 \neq c_3, c_2 \neq c_3 \\ ?-ALMENO\_TRE(x)$$

### Esercizio 5

Trovare gli studenti che hanno sostenuto esattamente due esami di corsi diversi

### Calcolo Relazionale

$$\{t|\exists t_1 \in ESAME, \exists t_2 \in ESAME(t[matr] = t_1[matr] \wedge t_1[matr] = t_2[matr] \wedge t_1[cod\_corso] \neq t_2[cod\_corso] \wedge \neg(\exists t_3 \in ESAME(t_1[matr] = t_3[matr] \wedge t_1[cod\_corso] \neq t_3[cod\_corso] \wedge t_2[cod\_corso] \neq t_3[cod\_corso])))\}$$

### Datalog

$$ALMENODUE(m) : -ESAME(m, c_1, -, -), ESAME(m, c_2, -, -), c_1 \neq c_2 \\ ALMENOTRE(m) : -ESAME(m, c_1, -, -), ESAME(m, c_2, -, -), ESAME(m, c_3, -, -), \\ c_1 \neq c_2, c_1 \neq c_3, c_2 \neq c_3 \\ ESATTAMENTE\_DUE(m) : -ALMENODUE(m), \neg ALMENOTRE(m) \\ ?-ESATTAMENTE\_DUE(x)$$

Non possiamo scrivere questa query in Datalog con una sola regola: ricordate la regola di safety per cui tutte le variabili contenuto del predicato che neghiamo devono comparire una altra volta in almeno un altro predicato nel corpo della regola stessa! Quindi una regola del tipo (come qualcuno ha proposto):

$$ALMENOTRE(m) : -ESAME(m, c_1, -, -), ESAME(m, c_2, -, -), \neg ESAME(m, c_3, -, -), \\ c_1 \neq c_2, c_1 \neq c_3, c_2 \neq c_3$$

NON è SAFE ( $c_3$  compare solo negato) è e un grave errore!!

### Esercizio 6

Trovare i distributori che non hanno fatto nessuna vendita successiva al loro ultimo rifornimento.

$$\pi_{CodiceDistributore} VENDITA - \pi_{CodiceDistributore} ((\pi_{CodDistributore, Data, Ora} VENDITA) \\ \bowtie_{CodDistributore=Codice \wedge (Data > DataUltimoRif \vee (Data=DataUltimoRif \wedge Ora > OraUltimoRif))} \\ (\pi_{Codice, DataUltimoRif, OraUltimoRif} DISTRIBUTORE)))$$

### Esercizio 7

Trovare l'ultima vendita effettuata di ciascun distributore.

$$(\pi_{CodiceDistributore, Data, Ora} VENDITA) - \\ ((\pi_{CodiceDistributore, Data, Ora} VENDITA) \bowtie_{CodiceDistributore=C \wedge (Data < D \vee (Data=D \wedge Ora < O))} \pi_{C, D, O} \\ (\rho_{(C, D, O) \leftarrow (CodiceDistributore, Data, Ora)} VENDITA)))$$

### Esercizio 8

Trovare le bevande che sono state vendute almeno due volte oppure che hanno un prezzo maggiore di 2 euro.

$$\pi_{CodBevanda} ( \\ \pi_{CodDistributore, Data, Ora, CodBevanda} VENDITA \\ \bowtie_{CodBevanda=CB \wedge (CodDistributore \neq CD \vee Data \neq D \vee Ora \neq O)} \\ \rho_{(CB, CD, D, O) \leftarrow (CodBevanda, CodDistributore, Data, Ora)} \pi_{CodDistributore, Data, Ora, CodBevanda} VENDITA) \\ \cup \\ \rho_{CodBevanda \leftarrow Codice} \pi_{Codice} (\sigma_{Prezzo > 2} BEVANDA)$$

### Esercizio 9

Trovare la bevanda più costosa.

$$\pi_{Codice} BEVANDA - \\ (\pi_{Codice, Prezzo} BEVANDA \bowtie_{Codice \neq C, Prezzo < P} (\rho_{(C, P) \leftarrow (Codice, Prezzo)} (\pi_{Codice, Prezzo} BEVANDA))))$$

### Esercizio 10

Trovare i distributori che hanno venduto tutte le bevande.

$$\pi_{CodDistributore} VENDITA - \\ \pi_{CodDistributore} ( \\ \pi_{CodDistributore, CodBevanda} (\pi_{CodDistributore} VENDITA \times \rho_{CodBevanda \leftarrow Codice} BEVANDA) - \\ \pi_{CodDistributore, CodBevanda} VENDITA)$$