

Dato lo schema:

PERSONA(id, nome, cognome, provincia_residenza, stato_civile)
ORGANIZZATORE(id, idevento)
PARTECIPANTE(idevento, idpersona)
CATERING(idcompagnia, nome, descrizione, sede)
CATEGORIAEVENTO(id, descrizione)
EVENTO(id, titolo, data, categoria, costo_partecipazione, idcatering)

Per risolvere le query useremo le relazioni con un nome abbreviato.

E ← EVENTO

O ← ORGANIZZATORE

P ← PERSONA

PA ← PARTECIPANTE

C ← CATERING

CE ← CATEGORIAEVENTO

1. Scrivere in algebra relazionale le seguenti query.

- a. Trovare nome e cognome dei partecipanti agli eventi che non si sono tenuti tenuti tra il 15 e il 20 ottobre 2009

$$\pi_{nome, cognome}(P \triangleright \triangleleft_{id=p.idpersona} PA \triangleright \triangleleft_{idevento=id} \sigma_{data \leq 15/10/2009 \vee data \geq 20/10/2013}(E))$$

- b. Trovare le persone che non hanno mai organizzato un evento di tipo "Compleanno".

$$\pi_{id}(O) - \pi_{O.id}(O \triangleright \triangleleft_{O.idevento=E.id} E \triangleright \triangleleft_{E.categoria=CE.id} \sigma_{descrizione='COMPLEANNO'}(CE))$$

- c. Trovare le persone che hanno partecipato a tutti gli eventi del 2013.

$$\delta_{id \leftarrow idevento}(PA) \div \pi_{id}(\sigma_{data \geq 1/1/2013 \wedge data \leq 31/12/2013}(E))$$

1. Scrivere in SQL le seguenti query:

- a. Trovare gli eventi che hanno avuto il maggiore incasso.

```
SELECT sum(costo_partecipazione) incasso, EVENTO.id
FROM EVENTO, PARTECIPANTE
WHERE EVENTO.id = PARTECIPANTE.idevento
group by EVENTO.ID
HAVING incasso >= ALL
      (SELECT sum(costo_partecipazione)
       FROM EVENTO, PARTECIPANTE
       WHERE EVENTO.id = PARTECIPANTE.idevento
       GROUP BY EVENTO.ID )
```

- b. Trovare gli eventi che hanno avuto il minor numero di partecipanti nel 2013.

```
SELECT count(*) num_partecipanti, EVENTO.id
FROM EVENTO, PARTECIPANTE
WHERE EVENTO.id=PARTECIPANTE.idevento
AND year(EVENTO.data) = 2013
GROUP BY EVENTO.id
HAVING count(*) <=ALL
      (SELECT count(*) num_partecipanti
       FROM EVENTO, PARTECIPANTE
       WHERE EVENTO.id=PARTECIPANTE.idevento
       AND year(EVENTO.data) = 2013
       GROUP BY EVENTO.id)
```

- c. Trovare le persone che hanno partecipato a tutti gli eventi organizzati da un organizzatore di Catania [risolvibile dopo la lezione di lunedì 9 novembre].

```
SELECT *
FROM persone p
WHERE NOT EXISTS (SELECT *
                  FROM organizzatore o, persona
                  o.id = persona.id AND
                  persona.provincia_residenza = 'CT' AND
                  NOT EXISTS (SELECT *
                             FROM partecipante pa
                             WHERE pa.idpersona=p.id AND
                                    pa.idevento=o.idevento))
```

- d. Supponendo che le coppie di persone che hanno partecipato allo stesso evento siano amici. Trovare le coppie di persone che non sono amiche ma sono connesse da un amico comune

```
CREATE VIEW amici AS
SELECT p1.idpersona as amico1, p2.idpersona as amico2, evento
FROM partecipante p1, partecipante p2
WHERE p1.idpersona > p2.idpersona
AND p1.idevento = p2.idevento

SELECT amico1, amico2
FROM amici
UNION
SELECT a1.amico1, a2.amico2
FROM amici a1, amici a2
WHERE a1.amico2 = a2.amico1
```

- e. Trovare gli eventi "Laurea" che hanno avuto il maggior numero di partecipanti "sposati".

```
SELECT count(*) num_partecipanti, Evento.id
FROM EVENTO, PARTECIPANTE
WHERE EVENTO.id=PARTECIPANTE.idevento
AND stato_civile="coniugato"
GROUP BY EVENTO.id
HAVING count(*) >=ALL
      (SELECT count(*)
       FROM EVENTO, PARTECIPANTE
       WHERE EVENTO.id=PARTECIPANTE.idevento
       AND stato_civile="coniugato"
       GROUP BY EVENTO.id)
```