

Basi di dati I — 5 settembre 2017
Tempo a disposizione: un'ora e 45 minuti.

Possibili soluzioni

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (15%) Considerare le seguenti relazioni (tutte senza valori nulli)

- PRODOTTI (Codice, Nome, Prezzo)
- MAGAZZINI (Codice, Nome, Città)
- GIACENZE (Prodotto, Magazzino, Quantità), con vincoli di integrità referenziale fra Prodotto e la chiave di PRODOTTI e fra Magazzino e la chiave di MAGAZZINI
- ORDINI (Codice, Prodotto, Quantità), con vincolo di integrità referenziale fra Prodotto e la chiave di PRODOTTI

Siano le seguenti le cardinalità delle relazioni

- PRODOTTI: cardinalità $P = 1500$
- MAGAZZINI: cardinalità $M = 20$
- GIACENZE: cardinalità $G = 20.000$
- ORDINI: cardinalità $O = 4000$

Indicare la cardinalità del risultato di ciascuna delle seguenti interrogazioni SQL, specificando l'intervallo nel quale essa può variare; indicare simboli e numeri.

	Min (simboli e valore)	Max (simboli e valore)
<pre>select prodotto, magazzino from giacenze join prodotti p on prodotto = p.codice where prezzo > 100</pre>	0 0	G 20.000
<pre>select prodotto, magazzino from giacenze join prodotti p on prodotto = p.codice join magazzini m on magazzino = m.codice where prezzo > 100</pre>	0 0	G 20.000
<pre>select o.codice as ordine, prodotto, nome, quantita from ordini o join prodotti p on prodotto = p.codice where quantita > 5</pre>	0 0	O 4000
<pre>select p.codice as prodotto, m.codice as magazzino from prodotti p, magazzini m</pre>	$P \times M$ 30.000	$P \times M$ 30.000

Basi di dati I — 5 settembre 2017

Domanda 2 (30%) Considerare la seguente base di dati su parte dello schema mostrato nella domanda precedente:

PRODOTTI			GIACENZE			ORDINI		
Codice	Nome	Prezzo	Prodotto	Magazzino	Quantità	Codice	Prodotto	Quantità
1	Frigo 12	1000	1	11	3	101	1	2
2	Lavatutto	800	1	12	2	102	2	2
3	Stirabene	50	2	11	1	103	2	1
4	Pila Stilo	1	3	11	10	104	3	1
			4	11	100	105	4	500
			4	12	200	106	4	300
			4	13	150			
			4	14	150			

Mostrare il risultato di ciascuna delle interrogazioni seguenti

1. `select codice, nome, sum(quantita) as giacenzatotale
from prodotti join giacenze on codice=prodotto
group by codice, nome`

codice integer	nome character(10)	giacenzatotale bigint
4	Pila Stilo	600
3	Stirabene	10
2	Lavatutto	1
1	Frigo 12	5

2. `select p.codice, nome, sum(quantita) as quantitaordinata
from prodotti p join ordini o on p.codice=prodotto
group by p.codice, nome`

codice integer	nome character(10)	quantitaordinata bigint
4	Pila Stilo	800
3	Stirabene	1
2	Lavatutto	3
1	Frigo 12	2

```

3. select p.codice, p.nome,
       sum(g.quantita) as giacenzatotale, sum(o.quantita) as quantitaordinata
   from prodotti p join giacenze g on p.codice=g.prodotto
       join ordini o on p.codice=o.prodotto
  group by p.codice, p.nome

```

(Si noti che questa interrogazione restituisce un risultato diverso rispetto a quello probabilmente atteso)

codice integer	nome character	giacenzat... bigint	quantitao... bigint
4	Pila Stilo	1200	3200
3	Stirabene	10	1
2	Lavatutto	2	3
1	Frigo 12	5	4

Formulare, sempre in SQL,

4. l'interrogazione che restituisce, per ciascun prodotto, il codice, il nome, la giacenza totale e la quantità totale ordinata (cioè ciò che probabilmente si desiderava ottenere con l'interrogazione precedente)

Risultato desiderato:

codice integer	nome character	giacenzat... bigint	quantitao... bigint
4	Pila Stilo	600	800
1	Frigo 12	5	2
3	Stirabene	10	1
2	Lavatutto	1	3

```

create or replace view totaliordinati as
  select prodotto, sum(quantita) as quantitaordinata
  from ordini
  group by prodotto;

```

```

create or replace view giacenzetotali as
  select prodotto, sum(quantita) as giacenzatotale
  from giacenze
  group by prodotto;

```

```

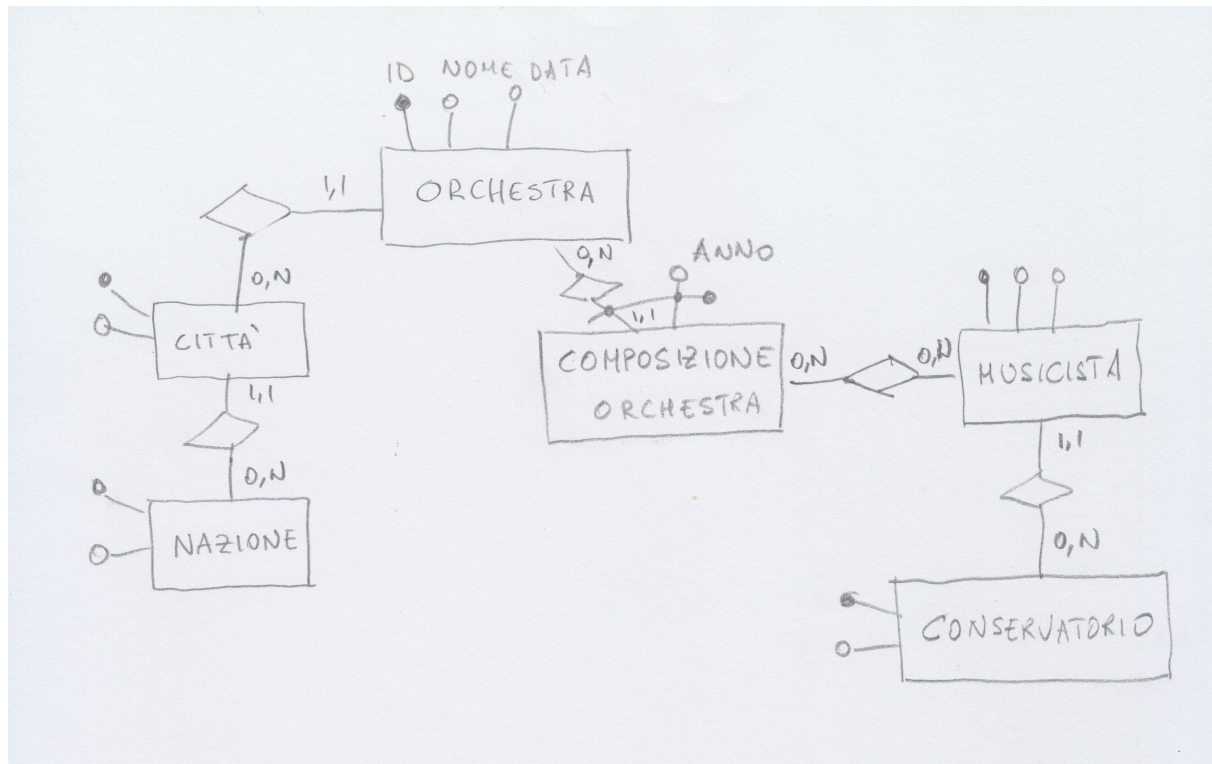
select p.codice, p.nome, g.giacenzatotale, o.quantitaordinata
  from prodotti p join totaliordinati o on p.codice=o.prodotto
       join giacenzetotali g on p.codice=g.prodotto

```

Domanda 3 (25%)

Definire uno schema Entity-Relationship che descriva i dati di interesse per un insieme di orchestre, secondo le seguenti specifiche:

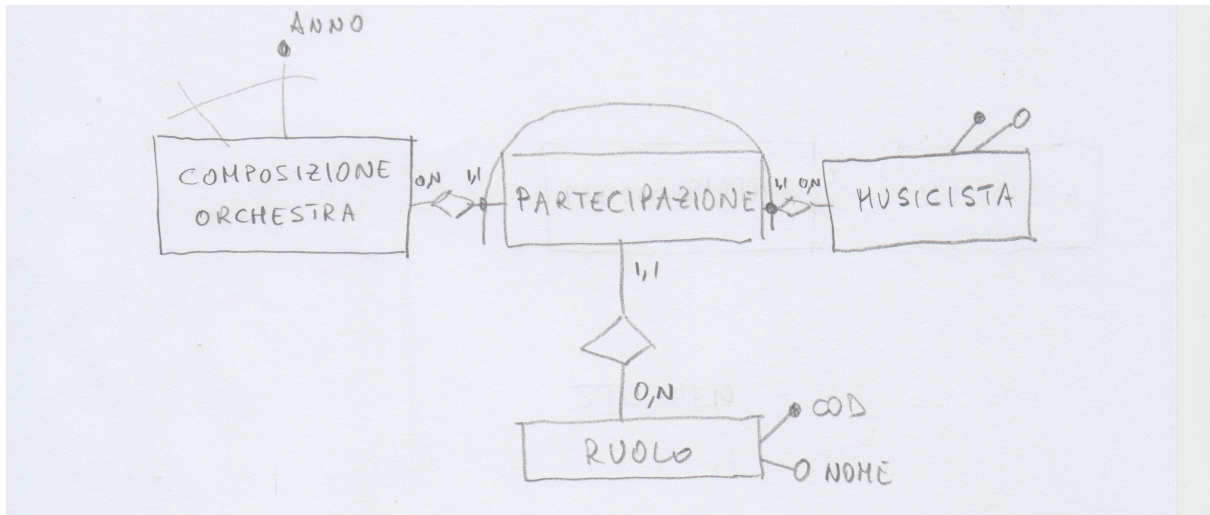
- ogni orchestra ha un nome, una sede (città e nazione, ciascuna con codice e nome) e una data di fondazione
- i componenti delle orchestre variano di anno in anno ed è importante tenere traccia della composizione in ciascun anno (ad esempio, sapere che Mario Rossi partecipa all'orchestra XY nel 2015 e nel 2017, ma non nel 2016); una stessa persona può partecipare a più orchestre, anche nello stesso anno
- ogni musicista (componente di orchestra), ha un codice identificativo, un cognome, un nome e si è diplomato presso un conservatorio
- ogni conservatorio ha un codice e un nome



Domanda 4 (15%)

Modificare lo schema proposto in risposta alla domanda precedente, per tenere conto delle seguenti specifiche aggiuntive (è sufficiente mostrare le porzioni di schema modificate, indicando come sono collegate al resto):

- ogni musicista, in ogni orchestra cui partecipa (in modo anche variabile da un anno all'altro), ha uno o più ruoli (pianista, primo violino, direttore,)
- i ruoli sono “standardizzati”, ciascuno con nome e codice (ad esempio il codice D corrisponde al direttore, il codice V1 al primo violino, etc.)



Basi di dati I — 5 settembre 2017

Domanda 5 (15%)

Considerare le seguenti quattro relazioni su uno stesso schema:

(A)				
FATTURE				
ID	Totale	IVA	Netto	OK
1	3000	800	2200	true
2	4000	1000	3000	true
3	3000	1000	2200	true

(B)				
FATTURE				
ID	Totale	IVA	Netto	OK
1	3000	800	2200	true
2	4000	1000	3000	true
3	3000	1000	2200	false

(C)				
FATTURE				
ID	Totale	IVA	Netto	OK
1	3000	800	2200	true
2	4000	1000	3000	false
3	3000	1000	2200	false

(D)				
FATTURE				
ID	Totale	IVA	Netto	OK
1	3000	800	2200	false
2	4000	1000	3000	false
3	3000	1000	2200	false

Considerare i tre vincoli di integrità mostrati nella tabella seguente e dire per ciascuno (con un sì o un no nelle celle corrispondenti), quali relazioni lo soddisfano e quali no:

	(A)	(B)	(C)	(D)
CHECK (((Netto = Totale - IVA) AND (OK = 'true')) OR ((Netto <> Totale - IVA) AND (OK = 'false')))	NO	SÌ	NO	NO
CHECK (NOT(Netto = Totale - IVA)) OR ((OK = 'true'))	SÌ	SÌ	NO	NO
CHECK ((NOT (OK = 'true')) OR (Netto = Totale - IVA))	NO	SÌ	SÌ	SÌ