

Domanda **1**

Risposta non ancora data

Punteggio max.: 1,00

Considerare le seguenti relazioni (tutte senza valori nulli)

- $R_1(A, B, C)$ , con vincolo di integrità referenziale fra  $B, C$  e la chiave  $D, E$  di  $R_2$  e con cardinalità  $C_1 = 1000$
- $R_2(D, E, F)$ , con vincolo di integrità referenziale fra  $F$  e la chiave di  $R_3$  e con cardinalità  $C_2 = 200$
- $R_3(G, H, I)$ , con cardinalità  $C_3 = 500$

Indicare la cardinalità del risultato di ciascuna delle seguenti espressioni (in cui il simbolo  $\wedge$  indica l'AND), specificando l'intervallo nel quale essa può variare; indicare simboli e numeri.

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
$(R_1 \bowtie_{(B=D)} R_2) \bowtie_{(F=G)} R_3$				
$R_1 \bowtie_{(B=D) \wedge (C=E)} R_2$				
$R_1 \bowtie_{(C=G)} R_3$				

Rispondere nella casella sottostante, su tre righe e quattro colonne, come avreste fatto nel modulo qui sopra

Paragrafo

Percorso: p

Domanda **2**

Risposta  
salvata

Punteggio  
max.: 1,00

Considerare le seguenti quattro relazioni su uno stesso schema:

(A)

STIPENDI				
ID	Lordo	Imposte	StipNetto	Verifica
1	3000	800	2200	true
2	4000	1000	3000	true
3	3000	1000	2200	true

(B)

STIPENDI				
ID	Lordo	Imposte	StipNetto	Verifica
1	3000	800	2200	true
2	4000	1000	3000	true
3	3000	1000	2200	false

(C)

STIPENDI				
ID	Lordo	Imposte	StipNetto	Verifica
1	3000	800	2200	true
2	4000	1000	3000	false
3	3000	1000	2200	false

(D)

STIPENDI				
ID	Lordo	Imposte	StipNetto	Verifica
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, quali relazioni lo soddisfano e quali no:

1. CHECK ( ( ( StipNetto = Lordo - Imposte) AND (Verifica = 'true') )  
OR ( (StipNetto <> Lordo - Imposte) AND (Verifica = 'false' ) ) )
2. CHECK ( NOT( StipNetto = Lordo - Imposte ) OR (Verifica = 'true') )
3. CHECK ( ( NOT (Verifica = 'true') ) OR ( StipNetto = Lordo - Imposte ) )

Rispondere su Moodle, su tre righe, su ogni riga quattro parole, ognuna “sì” o “no’.” Ad esempio (valori a caso):

sì sì no no  
no sì sì no  
sì no no no

Paragrafo

Percorso: p

Domanda **3**

Risposta  
incompleta

Punteggio  
max.: 1,00

Con riferimento ad una base di dati sullo schema

```
CREATE TABLE Compositori ( codice integer NOT NULL PRIMARY KEY,  
                           cognome text ,  
                           nome text );
```

```
CREATE TABLE Pezzi ( codice integer NOT NULL PRIMARY KEY,  
                     titolo text,  
                     autore integer NOT NULL REFERENCES Compositori,  
                     durata integer);
```

```
CREATE TABLE Concerti ( codice integer NOT NULL PRIMARY KEY,  
                        titolo text ,  
                        descrizione text );
```

```
CREATE TABLE Programmazione ( pezzo integer NOT NULL REFERENCES Pezzi,  
                               concerto integer NOT NULL REFERENCES Concerti,  
                               posizione integer,  
                               PRIMARY KEY(pezzo, concerto) );
```

Scrivere l'interrogazione SQL che trova i pezzi di durata maggiore di 5 il cui autore è un compositore di cognome 'Rossi'. Mostrare codice del compositore e codice, titolo e durata del pezzo. Ordinare per codice del compositore e codice del pezzo

**For example:**

Test	Result
-- Test con base di dati iniziale	<pre>codiceCompositore  codicePezzo  titolo  durata ----- 11                2          Ballata Y  8</pre>
-- Test con ennupla aggiunta	<pre>codiceCompositore  codicePezzo  titolo  durata ----- 11                2          Ballata Y  8 21                9          Sinfonia K 21</pre>

**Answer:** (penalty regime: 10, 20, ... %)

1 ||

Please provide a non-empty answer

Verifica risposta

	Test	Expected	Got	
✘	-- Test con base di dati iniziale	<pre> codiceCompositore  codicePezzo titolo             durata ----- 11                 2 Ballata Y         8 </pre>	<pre> codice  titolo duratatotale ----- 21      Bla 20 </pre>	✘
✘	-- Test con ennupla aggiunta	<pre> codiceCompositore  codicePezzo titolo             durata ----- 11                 2 Ballata Y         8 21                 9 Sinfonia K        21 </pre>	<pre> codice  titolo duratatotale ----- 21      Bla 20 </pre>	✘

Your code must pass all tests to earn any marks. Try again.

Show differences

La risposta non è valida, è stata ignorata senza dar luogo a penalità.

Domanda **4**

Risposta  
corretta

Punteggio  
max.: 1,00

Con riferimento ad una base di dati sullo schema

```
CREATE TABLE Compositori ( codice integer NOT NULL PRIMARY KEY,  
                             cognome text ,  
                             nome text );
```

```
CREATE TABLE Pezzi ( codice integer NOT NULL PRIMARY KEY,  
                      titolo text,  
                      autore integer NOT NULL REFERENCES Compositori,  
                      durata integer);
```

```
CREATE TABLE Concerti ( codice integer NOT NULL PRIMARY KEY,  
                         titolo text ,  
                         descrizione text );
```

```
CREATE TABLE Programmazione ( pezzo integer NOT NULL REFERENCES Pezzi,  
                               concerto integer NOT NULL REFERENCES Concerti,  
                               posizione integer,  
                               PRIMARY KEY(pezzo, concerto) );
```

Scrivere l'interrogazione SQL che trova codice e titolo dei pezzi che non sono presenti in nessun concerto. Ordinare per codice

**Answer:** (penalty regime: 10, 20, ... %)

```
1 SELECT DISTINCT Pezzi.codice, Pezzi.titolo FROM Pezzi,  
2 WHERE Pezzi.codice NOT IN (  
3     SELECT pezzo FROM Programmazione  
4 )  
5 ORDER BY Pezzi.codice
```

Verifica risposta

	Test	Expected	Got																					
✓	-- Test con base di dati iniziale	<table border="1"><thead><tr><th>codice</th><th>titolo</th></tr></thead><tbody><tr><td>4</td><td>Sinfonia X</td></tr><tr><td>5</td><td>Sinfonia X</td></tr><tr><td>6</td><td>Sinfonia Z</td></tr></tbody></table>	codice	titolo	4	Sinfonia X	5	Sinfonia X	6	Sinfonia Z	<table border="1"><thead><tr><th>codice</th><th>titolo</th></tr></thead><tbody><tr><td>4</td><td>Sinfonia X</td></tr><tr><td>5</td><td>Sinfonia X</td></tr><tr><td>6</td><td>Sinfonia Z</td></tr></tbody></table>	codice	titolo	4	Sinfonia X	5	Sinfonia X	6	Sinfonia Z	✓				
codice	titolo																							
4	Sinfonia X																							
5	Sinfonia X																							
6	Sinfonia Z																							
codice	titolo																							
4	Sinfonia X																							
5	Sinfonia X																							
6	Sinfonia Z																							
✓	-- Test con ennupla aggiunta	<table border="1"><thead><tr><th>codice</th><th>titolo</th></tr></thead><tbody><tr><td>4</td><td>Sinfonia X</td></tr><tr><td>5</td><td>Sinfonia X</td></tr><tr><td>6</td><td>Sinfonia Z</td></tr><tr><td>7</td><td>Sonata Z</td></tr></tbody></table>	codice	titolo	4	Sinfonia X	5	Sinfonia X	6	Sinfonia Z	7	Sonata Z	<table border="1"><thead><tr><th>codice</th><th>titolo</th></tr></thead><tbody><tr><td>4</td><td>Sinfonia X</td></tr><tr><td>5</td><td>Sinfonia X</td></tr><tr><td>6</td><td>Sinfonia Z</td></tr><tr><td>7</td><td>Sonata Z</td></tr></tbody></table>	codice	titolo	4	Sinfonia X	5	Sinfonia X	6	Sinfonia Z	7	Sonata Z	✓
codice	titolo																							
4	Sinfonia X																							
5	Sinfonia X																							
6	Sinfonia Z																							
7	Sonata Z																							
codice	titolo																							
4	Sinfonia X																							
5	Sinfonia X																							
6	Sinfonia Z																							
7	Sonata Z																							

Passed all tests! ✓

Domanda **5**

Risposta errata

Punteggio  
max.: 1,00

Con riferimento ad una base di dati sullo schema

```
CREATE TABLE Compositori ( codice integer NOT NULL PRIMARY KEY,  
                           cognome text ,  
                           nome text );
```

```
CREATE TABLE Pezzi ( codice integer NOT NULL PRIMARY KEY,  
                     titolo text,  
                     autore integer NOT NULL REFERENCES Compositori,  
                     durata integer);
```

```
CREATE TABLE Concerti ( codice integer NOT NULL PRIMARY KEY,  
                         titolo text ,  
                         descrizione text );
```

```
CREATE TABLE Programmazione ( pezzo integer NOT NULL REFERENCES Pezzi,  
                               concerto integer NOT NULL REFERENCES Concerti,  
                               posizione integer,  
                               PRIMARY KEY(pezzo, concerto) );
```

Scrivere l'interrogazione SQL che trova, per ogni compositore, il numero di pezzi diversi suonati in almeno un concerto. Trascurare i compositori per i quali non è suonato alcun pezzo. Mostrare codice, cognome, nome e numero di pezzi.

**For example:**

Test	Result												
-- Test con base di dati iniziale	<table><thead><tr><th>codice</th><th>cognome</th><th>nome</th><th>numeroPezzi</th></tr></thead><tbody><tr><td>11</td><td>Rossi</td><td>Mario</td><td>2</td></tr><tr><td>12</td><td>Bruni</td><td>Luca</td><td>1</td></tr></tbody></table>	codice	cognome	nome	numeroPezzi	11	Rossi	Mario	2	12	Bruni	Luca	1
codice	cognome	nome	numeroPezzi										
11	Rossi	Mario	2										
12	Bruni	Luca	1										
-- Test con ennupla aggiunta	<table><thead><tr><th>codice</th><th>cognome</th><th>nome</th><th>numeroPezzi</th></tr></thead><tbody><tr><td>11</td><td>Rossi</td><td>Mario</td><td>2</td></tr><tr><td>12</td><td>Bruni</td><td>Luca</td><td>2</td></tr></tbody></table>	codice	cognome	nome	numeroPezzi	11	Rossi	Mario	2	12	Bruni	Luca	2
codice	cognome	nome	numeroPezzi										
11	Rossi	Mario	2										
12	Bruni	Luca	2										

**Answer:** (penalty regime: 10, 20, ... %)

```
1 SELECT Compositori.codice, Compositori.cognome, Compos  
2     SELECT COUNT(Pezzi.codice) FROM Pezzi, Compositori  
3     WHERE Pezzi.autore = Compositori.codice  
4     GROUP BY Compositori.codice  
5 ) AS numeroPezzi FROM Compositori, Pezzi  
6 WHERE numeroPezzi > 0 AND Compositori.codice = Pezzi.a  
7 GROUP BY Compositori.codice  
8
```

Verifica risposta

	Test	Expected	Got	
✘	-- Test con base di dati iniziale	<pre> codice    cognome   nome numeroPezzi ----- 11        Rossi     Mario 2 12        Bruni     Luca 1 </pre>	<pre> codice    cognome   nome numeroPezzi ----- 11        Rossi     Mario 2 12        Bruni     Luca 2 13        Neri      Gino 2 </pre>	✘
✘	-- Test con ennupla aggiunta	<pre> codice    cognome   nome numeroPezzi ----- 11        Rossi     Mario 2 12        Bruni     Luca 2 </pre>	<pre> codice    cognome   nome numeroPezzi ----- 11        Rossi     Mario 2 12        Bruni     Luca 2 13        Neri      Gino 2 </pre>	✘

Your code must pass all tests to earn any marks. Try again.

Show differences

Domanda **6**

Risposta  
mancante

Punteggio  
max.: 1,00

Con riferimento ad una base di dati sullo schema

```
CREATE TABLE Compositori ( codice integer NOT NULL PRIMARY KEY,  
                           cognome text ,  
                           nome text );
```

```
CREATE TABLE Pezzi ( codice integer NOT NULL PRIMARY KEY,  
                     titolo text,  
                     autore integer NOT NULL REFERENCES Compositori,  
                     durata integer);
```

```
CREATE TABLE Concerti ( codice integer NOT NULL PRIMARY KEY,  
                         titolo text ,  
                         descrizione text );
```

```
CREATE TABLE Programmazione ( pezzo integer NOT NULL REFERENCES Pezzi,  
                               concerto integer NOT NULL REFERENCES Concerti,  
                               posizione integer,  
                               PRIMARY KEY(pezzo, concerto) );
```

Scrivere l'interrogazione SQL che trova il compositore (o i compositori) con il massimo numero di pezzi diversi suonati in almeno un concerto (come calcolati in risposta alla domanda precedente). Mostrare codice, nome, cognome e numero di pezzi e ordinare per codice.

**For example:**

Test	Result												
-- Test con base di dati iniziale	<table><thead><tr><th>codice</th><th>cognome</th><th>nome</th><th>numeroPezzi</th></tr></thead><tbody><tr><td>11</td><td>Rossi</td><td>Mario</td><td>2</td></tr></tbody></table>	codice	cognome	nome	numeroPezzi	11	Rossi	Mario	2				
codice	cognome	nome	numeroPezzi										
11	Rossi	Mario	2										
-- Test con ennupla aggiunta	<table><thead><tr><th>codice</th><th>cognome</th><th>nome</th><th>numeroPezzi</th></tr></thead><tbody><tr><td>11</td><td>Rossi</td><td>Mario</td><td>2</td></tr><tr><td>12</td><td>Bruni</td><td>Luca</td><td>2</td></tr></tbody></table>	codice	cognome	nome	numeroPezzi	11	Rossi	Mario	2	12	Bruni	Luca	2
codice	cognome	nome	numeroPezzi										
11	Rossi	Mario	2										
12	Bruni	Luca	2										

**Answer:** (penalty regime: 10, 20, ... %)

1 ||

Verifica risposta

◀ Esercizi

Vai a...

Turno 1 bis (nascosto) ▶