

Basi di dati I — 10 luglio 2017
Tempo a disposizione: un'ora e 30 minuti.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (20%)

Considerare le seguenti quattro relazioni su uno stesso schema:

<p>(A)</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th colspan="5">STIPENDI</th> </tr> <tr> <th>ID</th> <th>Lordo</th> <th>Tasse</th> <th>Netto</th> <th>OK</th> </tr> </thead> <tbody> <tr><td>1</td><td>3000</td><td>800</td><td>2200</td><td>true</td></tr> <tr><td>2</td><td>4000</td><td>1000</td><td>3000</td><td>true</td></tr> <tr><td>3</td><td>3000</td><td>1000</td><td>2200</td><td>true</td></tr> </tbody> </table>	STIPENDI					ID	Lordo	Tasse	Netto	OK	1	3000	800	2200	true	2	4000	1000	3000	true	3	3000	1000	2200	true	<p>(B)</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th colspan="5">STIPENDI</th> </tr> <tr> <th>ID</th> <th>Lordo</th> <th>Tasse</th> <th>Netto</th> <th>OK</th> </tr> </thead> <tbody> <tr><td>1</td><td>3000</td><td>800</td><td>2200</td><td>true</td></tr> <tr><td>2</td><td>4000</td><td>1000</td><td>3000</td><td>true</td></tr> <tr><td>3</td><td>3000</td><td>1000</td><td>2200</td><td>false</td></tr> </tbody> </table>	STIPENDI					ID	Lordo	Tasse	Netto	OK	1	3000	800	2200	true	2	4000	1000	3000	true	3	3000	1000	2200	false
STIPENDI																																																			
ID	Lordo	Tasse	Netto	OK																																															
1	3000	800	2200	true																																															
2	4000	1000	3000	true																																															
3	3000	1000	2200	true																																															
STIPENDI																																																			
ID	Lordo	Tasse	Netto	OK																																															
1	3000	800	2200	true																																															
2	4000	1000	3000	true																																															
3	3000	1000	2200	false																																															
<p>(C)</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th colspan="5">STIPENDI</th> </tr> <tr> <th>ID</th> <th>Lordo</th> <th>Tasse</th> <th>Netto</th> <th>OK</th> </tr> </thead> <tbody> <tr><td>1</td><td>3000</td><td>800</td><td>2200</td><td>true</td></tr> <tr><td>2</td><td>4000</td><td>1000</td><td>3000</td><td>false</td></tr> <tr><td>3</td><td>3000</td><td>1000</td><td>2200</td><td>false</td></tr> </tbody> </table>	STIPENDI					ID	Lordo	Tasse	Netto	OK	1	3000	800	2200	true	2	4000	1000	3000	false	3	3000	1000	2200	false	<p>(D)</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th colspan="5">STIPENDI</th> </tr> <tr> <th>ID</th> <th>Lordo</th> <th>Tasse</th> <th>Netto</th> <th>OK</th> </tr> </thead> <tbody> <tr><td>1</td><td>3000</td><td>800</td><td>2200</td><td>false</td></tr> <tr><td>2</td><td>4000</td><td>1000</td><td>3000</td><td>false</td></tr> <tr><td>3</td><td>3000</td><td>1000</td><td>2200</td><td>false</td></tr> </tbody> </table>	STIPENDI					ID	Lordo	Tasse	Netto	OK	1	3000	800	2200	false	2	4000	1000	3000	false	3	3000	1000	2200	false
STIPENDI																																																			
ID	Lordo	Tasse	Netto	OK																																															
1	3000	800	2200	true																																															
2	4000	1000	3000	false																																															
3	3000	1000	2200	false																																															
STIPENDI																																																			
ID	Lordo	Tasse	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 = Lordo - Tasse) AND (OK = 'true')) OR ((Netto <> Lordo - Tasse) AND (OK = 'false')))				
CHECK ((NOT (OK = 'true')) OR (Netto = Lordo - Tasse))				
CHECK (NOT(Netto = Lordo - Tasse)) OR (((OK = 'true'))				

Domanda 2 (10%)

Con riferimento ad una relazione **GIOCATORI**(Codice, Nome, Altezza, Ruolo), scrivere le interrogazioni SQL che calcolano l'altezza media dei giocatori di ciascun ruolo, nei due casi seguenti:

- si usa il valore nullo per indicare che l'altezza non è nota

- si usa il valore 0 per indicare che l'altezza non è nota

Domanda 3 (25%)

Definire uno schema Entity-Relationship che descriva i dati di interesse per una biblioteca, secondo le seguenti specifiche:

- oggetto dei prestiti sono esemplari (detti anche volumi o copie) di libri, identificati attraverso un numero di inventario; ognuno ha una data di acquisto;
- ogni libro
 - può essere presente in più copie
 - ha un codice (chiamato ISBN),
 - un titolo
 - uno o più autori
 - è pubblicato da una casa editrice
- ogni casa editrice ha un nome e un codice;
- ogni autore ha un codice, un cognome, un nome e una data di nascita e una affiliazione (un “ente”, con codice e nome)
- per ogni prestito in corso (quelli conclusi non interessano), sono rilevanti la data prevista di restituzione e l’utente (che può avere più volumi in prestito contemporaneamente), con codice identificativo, nome, cognome e recapito telefonico.

Domanda 4 (20%)

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):

- interessano anche i prestiti conclusi, con la data di restituzione; notare che uno stesso utente può avere preso in prestito più volte lo stesso libro;
- un autore può avere (nel tempo) più affiliazioni ed è di interesse associare ad ogni libro una (e una sola) delle affiliazioni (nota bene: non è importante tenere traccia dell'evoluzione temporale delle affiliazioni, ma solo dell'affiliazione associata a ciascun libro).

Domanda 5 (25%)

Considerare la seguente base di dati relazionale (semplificazione di quella dell'esercitazione realizzativa discussa durante il corso; si noti che i valori nulli per l'attributo **DataPagamento** indicano che il pagamento non è stato effettuato; tutti gli altri attributi non ammettono valore nullo):

MODULI			
ID	Contribuente	Saldo	DataPagamento
10	RSSMRA1	1.100	25/06/2016
11	RSSMRA1	1.100	25/10/2016
12	RSSMRA1	1.600	NULL
13	BNCPLA1	400	25/10/2016
14	BRNPLA1	1.200	27/10/2016

RIGHE				
Modulo	Riga	Tributo	Anno	Importo
10	1	IRPEF	2016	1.000
10	2	IMU	2016	100
11	1	IRPEF	2016	1.000
11	2	IMU	2016	100
12	1	IRPEF	2017	1.500
12	2	IMU	2017	100
13	1	IVA	2015	200
13	2	IMU	2016	200
14	1	IRPEF	2015	1.200

CONTRIBUENTI		
CF	Cognome	Nome
RSSMRA1	Rossi	Mario
BNCPLA1	Bianchi	Paolo
VRDPLA1	Verdi	Paola
BRNPLA1	Bruni	Paolo

1. formulare in SQL l'interrogazione che trova i tributi (con i relativi importi complessivi) pagati dal contribuente il cui codice fiscale è "RSSMRA1" (per comodità viene riportato il risultato rispetto alla base di dati mostrata sopra):

Tributo	Totale
IRPEF	2.000
IMU	200

2. formulare in SQL l'interrogazione che trova, per ciascun contribuente (mostrando CF, cognome e nome) il totale degli importi dovuti per l'IRPEF, di quelli pagati e di quelli dovuti e non ancora pagati

CF	Cognome	Nome	dovuta	pagata	dapagare
RSSMRA1	Rossi	Mario	3.500	2.000	1.500
BRNPLA1	Bruni	Paolo	1.200	1.200	0

Basi di dati I — 10 luglio 2017
Tempo a disposizione: un'ora e 30 minuti.

Possibili soluzioni

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (20%)

Considerare le seguenti quattro relazioni su uno stesso schema:

(A)					(B)				
STIPENDI					STIPENDI				
ID	Lordo	Tasse	Netto	OK	ID	Lordo	Tasse	Netto	OK
1	3000	800	2200	true	1	3000	800	2200	true
2	4000	1000	3000	true	2	4000	1000	3000	true
3	3000	1000	2200	true	3	3000	1000	2200	false

(C)					(D)				
STIPENDI					STIPENDI				
ID	Lordo	Tasse	Netto	OK	ID	Lordo	Tasse	Netto	OK
1	3000	800	2200	true	1	3000	800	2200	false
2	4000	1000	3000	false	2	4000	1000	3000	false
3	3000	1000	2200	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 = Lordo - Tasse) AND (OK = 'true')) OR ((Netto <> Lordo - Tasse) AND (OK = 'false')))	NO	SÌ	NO	NO
CHECK ((NOT (OK = 'true')) OR (Netto = Lordo - Tasse))	NO	SÌ	SÌ	SÌ
CHECK (NOT(Netto = Lordo - Tasse)) OR (((OK = 'true'))	SÌ	SÌ	NO	NO

Domanda 2 (10%)

Con riferimento ad una relazione **GIOCATORI**(Codice, Nome, Altezza, Ruolo), scrivere le interrogazioni SQL che calcolano l'altezza media dei giocatori di ciascun ruolo, nei due casi seguenti:

- si usa il valore nullo per indicare che l'altezza non è nota

Soluzione

```
SELECT Ruolo , AVG(Altezza) AS AltezzaMedia
FROM Giocatori
GROUP BY Ruolo
```

- si usa il valore 0 per indicare che l'altezza non è nota

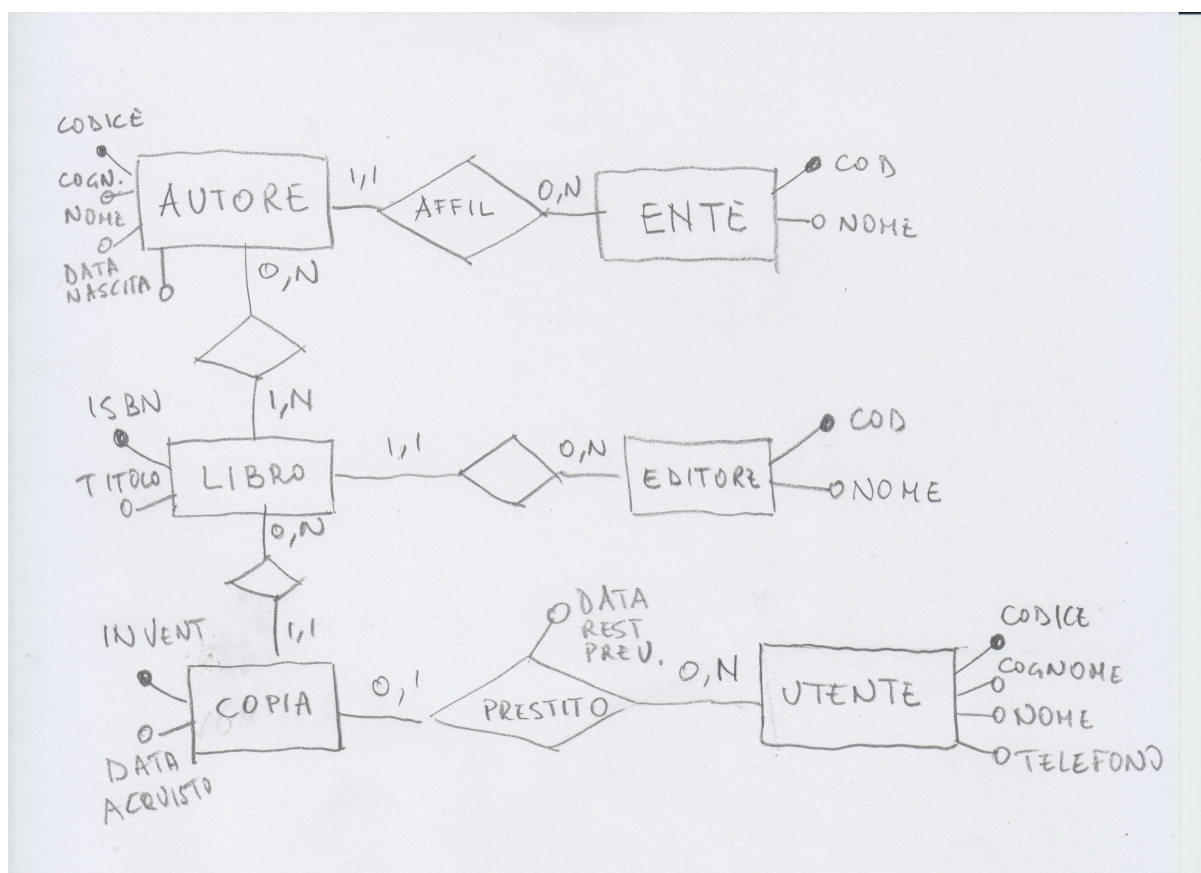
Soluzione

```
SELECT Ruolo , AVG(Altezza) AS AltezzaMedia
FROM Giocatori
WHERE Altezza <> 0
GROUP BY Ruolo
```

Domanda 3 (25%)

Definire uno schema Entity-Relationship che descriva i dati di interesse per una biblioteca, secondo le seguenti specifiche:

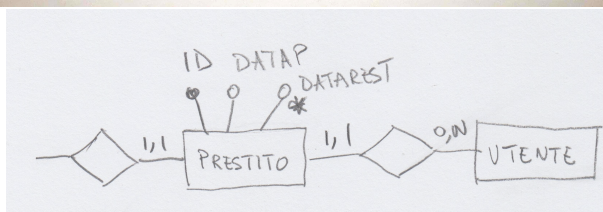
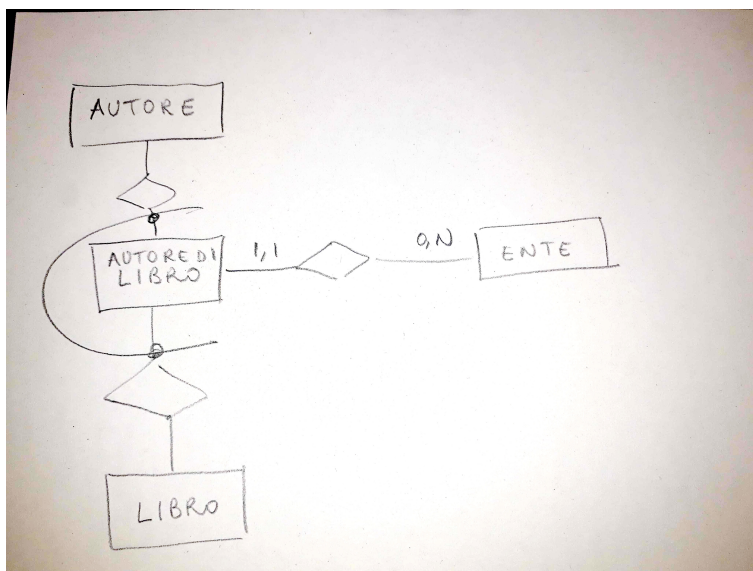
- oggetto dei prestiti sono esemplari (detti anche volumi o copie) di libri, identificati attraverso un numero di inventario; ognuno ha una data di acquisto;
- ogni libro
 - può essere presente in più copie
 - ha un codice (chiamato ISBN),
 - un titolo
 - uno o più autori
 - è pubblicato da una casa editrice
- ogni casa editrice ha un nome e un codice;
- ogni autore ha un codice, un cognome, un nome e una data di nascita e una affiliazione (un “ente”, con codice e nome)
- per ogni prestito in corso (quelli conclusi non interessano), sono rilevanti la data prevista di restituzione e l’utente (che può avere più volumi in prestito contemporaneamente), con codice identificativo, nome, cognome e recapito telefonico.



Domanda 4 (20%)

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):

- interessano anche i prestiti conclusi, con la data di restituzione; notare che uno stesso utente può avere preso in prestito più volte lo stesso libro;
- un autore può avere (nel tempo) più affiliazioni ed è di interesse associare ad ogni libro una (e una sola) delle affiliazioni (nota bene: non è importante tenere traccia dell'evoluzione temporale delle affiliazioni, ma solo dell'affiliazione associata a ciascun libro).



Domanda 5 (25%)

Considerare la seguente base di dati relazionale (semplificazione di quella dell'esercitazione realizzativa discussa durante il corso; si noti che i valori nulli per l'attributo `DataPagamento` indicano che il pagamento non è stato effettuato; tutti gli altri attributi non ammettono valore nullo):

MODULI			
ID	Contribuente	Saldo	DataPagamento
10	RSSMRA1	1.100	25/06/2016
11	RSSMRA1	1.100	25/10/2016
12	RSSMRA1	1.600	NULL
13	BNCPLA1	400	25/10/2016
14	BRNPLA1	1.200	27/10/2016

RIGHE				
Modulo	Riga	Tributo	Anno	Importo
10	1	IRPEF	2016	1.000
10	2	IMU	2016	100
11	1	IRPEF	2016	1.000
11	2	IMU	2016	100
12	1	IRPEF	2017	1.500
12	2	IMU	2017	100
13	1	IVA	2015	200
13	2	IMU	2016	200
14	1	IRPEF	2015	1.200

CONTRIBUENTI		
CF	Cognome	Nome
RSSMRA1	Rossi	Mario
BNCPLA1	Bianchi	Paolo
VRDPLA1	Verdi	Paola
BRNPLA1	Bruni	Paolo

- formulare in SQL l'interrogazione che trova i tributi (con i relativi importi complessivi) pagati dal contribuente il cui codice fiscale è "RSSMRA1" (per comodità viene riportato il risultato rispetto alla base di dati mostrata sopra):

Tributo	Totale
IRPEF	2.000
IMU	200

```

CREATE VIEW modulirighe AS
SELECT *
FROM moduli JOIN righe ON id = modulo;

SELECT Tributo, SUM(Importo) AS TOTALE
FROM modulirighe
WHERE Contribuente = 'RSSMRA1'
AND DataPagamento IS NOT NULL
GROUP BY Tributo;
    
```

- formulare in SQL l'interrogazione che trova, per ciascun contribuente (mostrando CF, cognome e nome) il totale degli importi dovuti per l'IRPEF, di quelli pagati e di quelli dovuti e non ancora pagati

CF	Cognome	Nome	dovuta	pagata	dapagare
RSSMRA1	Rossi	Mario	3.500	2.000	1.500
BRNPLA1	Bruni	Paolo	1.200	1.200	0

```

CREATE VIEW IRPEFDovuta
AS SELECT Contribuente AS CF, SUM(Importo) AS Dovuta
FROM ModuliRighe
WHERE Tributo = 'IRPEF'
GROUP BY Contribuente;

CREATE VIEW IRPEFPagata
AS SELECT Contribuente AS CF, SUM(Importo) AS Pagata
FROM ModuliRighe
WHERE Tributo = 'IRPEF' AND DataPagamento IS NOT NULL
GROUP BY Contribuente;

SELECT c.CF, Cognome, Nome, Dovuta, Pagata, Dovuta-Pagata AS DaPagare
FROM Contribuenti c JOIN IRPEFDovuta d ON c.CF=d.CF
JOIN IRPEFPagata p ON c.CF=p.CF
    
```

Questa interrogazione restituisce i contribuenti che hanno sia IRPEF dovuta sia IRPEF pagata; se ci fossero contribuenti con IRPEF dovuta e senza IRPEF pagata, servirebbe un `LEFT JOIN` e un meccanismo (`COALESCE`, che non abbiamo visto) per trasformare i valori nulli in interi pari a zero. Ma tutto ciò non era richiesto.