

Basi di dati — 18 novembre 2019 — Prova parziale — Compito A
Tempo a disposizione: un'ora.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (75%) Considerare il seguente schema relazionale

```
CREATE TABLE Persone ( CF text NOT NULL PRIMARY KEY,
                        cognome text,
                        nome text );
CREATE TABLE Ordini ( codice integer NOT NULL PRIMARY KEY,
                        cliente text NOT NULL REFERENCES persone,
                        descrizione text NOT NULL);
CREATE TABLE RateEmesse ( numero integer NOT NULL PRIMARY KEY,
                            ordine integer NOT NULL REFERENCES ordini ,
                            data date NOT NULL,
                            importo integer NOT NULL,
                            datapagamento date );
```

Si noti che ogni ordine è relativo ad un cliente e che ogni rata emessa è relativa ad un ordine.
Formulare la seguente interrogazione in algebra relazionale

1. Mostrare codice fiscale, nome e cognome delle persone che non hanno nessun ordine

Formulare le seguenti interrogazioni in SQL

2. Per ciascun ordine che abbia rate emesse, mostrare l'importo totale delle rate stesse

3. Per ciascuna persona, mostrare il numero di ordini, mostrando anche (con il valore 0) le persone che non hanno ordini.

Basi di dati I — 18 novembre 2019 — Compito A

4. Per ciascun ordine che abbia rate emesse, mostrare il totale delle rate emesse, il totale delle rate pagate e l'eventuale “debito”, cioè la differenza fra il totale delle rate emesse e il totale delle rate pagate. Per semplicità, per questa interrogazione, supporre che tutti gli ordini abbiano almeno una rata emessa e almeno una rata pagata.

5. Mostrare le informazioni sull'ordine per il quale è massimo il “debito” di cui all'interrogazione precedente

Domanda 2 (15%) Considerare le seguenti relazioni (tutte senza valori nulli)

- $R_1(\underline{A}, B)$, con vincolo di integrità referenziale fra B e la chiave D di R_2 e con cardinalità $C_1 = 1000$
- $R_2(\underline{D}, E, F, G)$, con vincolo di integrità referenziale fra F, G e la chiave H, P di R_3 e con cardinalità $C_2 = 400$
- $R_3(\underline{H}, \underline{P}, Q)$, 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)
$\pi_{EF}(R_2)$	1	C2		
$\min = \max = C1$ $(R_1 \bowtie_{(B=D)} R_2) \bowtie_{(F=H) \wedge (G=P)} R_3$	C1	C1		
$R_2 \bowtie_{(F=H) \wedge (G=P)} R_3$	C2	C2		
$R_1 \bowtie_{(B=P)} R_3$	0	C1*C3		

Domanda 3 (10%)

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

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

- si usa il valore -1 per indicare che l'altezza non è nota

Basi di dati — 18 novembre 2019 — Prova parziale — Compito B
Tempo a disposizione: un'ora.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (75%) Considerare il seguente schema relazionale

```
CREATE TABLE Persone ( CF text NOT NULL PRIMARY KEY,
                        cognome text,
                        nome text );
CREATE TABLE Ordini ( codice integer NOT NULL PRIMARY KEY,
                        cliente text NOT NULL REFERENCES persone,
                        descrizione text NOT NULL);
CREATE TABLE RateEmesse ( numero integer NOT NULL PRIMARY KEY,
                            ordine integer NOT NULL REFERENCES ordini ,
                            data date NOT NULL,
                            importo integer NOT NULL,
                            datapagamento date );
```

Si noti che ogni ordine è relativo ad un cliente e che ogni rata emessa è relativa ad un ordine.
Formulare la seguente interrogazione in algebra relazionale

1. Mostrare codice, cliente e descrizione degli ordini per i quali non c'è nessuna rata emessa

Formulare le seguenti interrogazioni in SQL

2. Per ciascuna persona che abbia ordini, mostrare il numero totale di tali ordini

3. Per ciascun ordine, mostrare l'importo totale delle rate emesse, mostrando anche (con il valore 0) gli ordini che non hanno rate emesse.

Basi di dati I — 18 novembre 2019 — Compito B

4. Per ciascun ordine che abbia rate emesse, mostrare il totale delle rate emesse, il totale delle rate pagate e l'eventuale “debito”, cioè la differenza fra il totale delle rate emesse e il totale delle rate pagate. Per semplicità, per questa interrogazione, supporre che tutti gli ordini abbiano almeno una rata emessa e almeno una rata pagata.

5. Mostrare le informazioni sull'ordine per il quale è massimo il “debito” di cui all'interrogazione precedente

Domanda 2 (15%) Considerare le seguenti relazioni (tutte senza valori nulli)

- $R_1(\underline{A}, B)$, con vincolo di integrità referenziale fra B e la chiave D di R_2 e con cardinalità $L_1 = 1000$
- $R_2(\underline{D}, E, F, G)$, con vincolo di integrità referenziale fra F, G e la chiave H, P di R_3 e con cardinalità $L_2 = 400$
- $R_3(\underline{H}, \underline{P}, Q)$, con cardinalità $L_3 = 500$

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)
$R_1 \bowtie_{(B=D)} R_2$	R1	R1		
$\min=\max=R1$ $(R_1 \bowtie_{(B=D)} R_2) \bowtie_{(F=H) \wedge (G=P)} R_3$	C1	C1		
$\pi_{HP}(R_3)$	R3	R3		
$R_3 \bowtie_{(Q=A)} R_1$	0	R3		

Domanda 3 (10%)

Con riferimento ad una relazione IMPIEGATI(Codice, Nome, Retribuzione, Livello), scrivere le interrogazioni SQL che calcolano, per ciascun livello, la retribuzione media degli impiegati di tale livello, nei due casi seguenti:

- si usa il valore nullo per indicare che la retribuzione non è nota

- si usa il valore 0 per indicare che la retribuzione non è nota

Basi di dati — 18 novembre 2019 — Prova parziale — Compito C
Tempo a disposizione: un'ora.

Cognome: _____ **Nome:** _____ **Matricola:** _____

Domanda 1 (75%) Considerare il seguente schema relazionale

```
CREATE TABLE Persone ( CF text NOT NULL PRIMARY KEY,
                        cognome text,
                        nome text );
CREATE TABLE Ordini ( codice integer NOT NULL PRIMARY KEY,
                        cliente text NOT NULL REFERENCES persone,
                        descrizione text NOT NULL);
CREATE TABLE RateEmesse ( numero integer NOT NULL PRIMARY KEY,
                            ordine integer NOT NULL REFERENCES ordini ,
                            data date NOT NULL,
                            importo integer NOT NULL,
                            datapagamento date );
```

Si noti che ogni ordine è relativo ad un cliente e che ogni rata emessa è relativa ad un ordine.
Formulare la seguente interrogazione in algebra relazionale

1. Mostrare codice fiscale, nome e cognome delle persone che non hanno nessun ordine

Formulare le seguenti interrogazioni in SQL

2. Per ciascun ordine che abbia rate emesse, mostrare l'importo totale delle rate stesse

3. Per ciascuna persona, mostrare il numero di ordini, mostrando anche (con il valore 0) le persone che non hanno ordini.

Basi di dati I — 18 novembre 2019 — Compito C

4. Per ciascun ordine che abbia rate emesse, mostrare il totale delle rate emesse, il totale delle rate pagate e l'eventuale “debito”, cioè la differenza fra il totale delle rate emesse e il totale delle rate pagate. Per semplicità, per questa interrogazione, supporre che tutti gli ordini abbiano almeno una rata emessa e almeno una rata pagata.

5. Mostrare le informazioni sull'ordine per il quale è massimo il “debito” di cui all'interrogazione precedente

Domanda 2 (15%) Considerare le seguenti relazioni (tutte senza valori nulli)

- $R_1(\underline{A}, B)$, con vincolo di integrità referenziale fra B e la chiave D di R_2 e con cardinalità $N_1 = 1000$
- $R_2(\underline{D}, E, F, G)$, con vincolo di integrità referenziale fra F, G e la chiave H, P di R_3 e con cardinalità $N_2 = 400$
- $R_3(\underline{H}, \underline{P}, Q)$, con cardinalità $N_3 = 500$

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)
$(R_1 \bowtie_{(B=D)} R_2) \bowtie_{(F=H) \wedge (G=P)} R_3$				
$R_2 \bowtie_{(F=H) \wedge (G=P)} R_3$				
$R_1 \bowtie_{(B=P)} R_3$				
$\pi_{EF}(R_2)$				

Domanda 3 (10%)

Con riferimento ad una relazione CALCIATORI(Codice, Nome, Altezza, Ruolo), scrivere le interrogazioni SQL che calcolano, per ciascun ruolo, l'altezza media dei calciatori di tale 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

Basi di dati — 18 novembre 2019 — Prova parziale — Compito D
Tempo a disposizione: un'ora.

Cognome: _____ **Nome:** _____ **Matricola:** _____

Domanda 1 (75%) Considerare il seguente schema relazionale

```
CREATE TABLE Persone ( CF text NOT NULL PRIMARY KEY,
                        cognome text,
                        nome text );
CREATE TABLE Ordini ( codice integer NOT NULL PRIMARY KEY,
                        cliente text NOT NULL REFERENCES persone,
                        descrizione text NOT NULL);
CREATE TABLE RateEmesse ( numero integer NOT NULL PRIMARY KEY,
                            ordine integer NOT NULL REFERENCES ordini ,
                            data date NOT NULL,
                            importo integer NOT NULL,
                            datapagamento date );
```

Si noti che ogni ordine è relativo ad un cliente e che ogni rata emessa è relativa ad un ordine.
Formulare la seguente interrogazione in algebra relazionale

1. Mostrare codice, cliente e descrizione degli ordini per i quali non c'è nessuna rata emessa

Formulare le seguenti interrogazioni in SQL

2. Per ciascuna persona che abbia ordini, mostrare il numero totale di tali ordini

3. Per ciascun ordine, mostrare l'importo totale delle rate emesse, mostrando anche (con il valore 0) gli ordini che non hanno rate emesse.

Basi di dati I — 18 novembre 2019 — Compito D

4. Per ciascun ordine che abbia rate emesse, mostrare il totale delle rate emesse, il totale delle rate pagate e l'eventuale “debito”, cioè la differenza fra il totale delle rate emesse e il totale delle rate pagate. Per semplicità, per questa interrogazione, supporre che tutti gli ordini abbiano almeno una rata emessa e almeno una rata pagata.

5. Mostrare le informazioni sull'ordine per il quale è massimo il “debito” di cui all'interrogazione precedente

Domanda 2 (15%) Considerare le seguenti relazioni (tutte senza valori nulli)

- $R_1(\underline{A}, B)$, con vincolo di integrità referenziale fra B e la chiave D di R_2 e con cardinalità $M_1 = 1000$
- $R_2(\underline{D}, E, F, G)$, con vincolo di integrità referenziale fra F, G e la chiave H, P di R_3 e con cardinalità $M_2 = 400$
- $R_3(\underline{H}, \underline{P}, Q)$, con cardinalità $M_3 = 500$

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)
$R_3 \bowtie_{(Q=A)} R_1$				
$\pi_{HP}(R_3)$				
$R_1 \bowtie_{(B=D)} R_2$				
$(R_1 \bowtie_{(B=D)} R_2) \bowtie_{(F=H) \wedge (G=P)} R_3$				

Domanda 3 (10%)

Con riferimento ad una relazione DIPENDENTI(Codice, Nome, Retribuzione, Livello), scrivere le interrogazioni SQL che calcolano, per ciascun livello, la retribuzione media dei dipendenti di tale livello, nei due casi seguenti:

- si usa il valore nullo per indicare che la retribuzione non è nota

- si usa il valore -1 per indicare che la retribuzione non è nota

Basi di dati — 18 novembre 2019 — Prova parziale — Compito A

Possibili soluzioni

Tempo a disposizione: un'ora.

Cognome: _____ **Nome:** _____ **Matricola:** _____

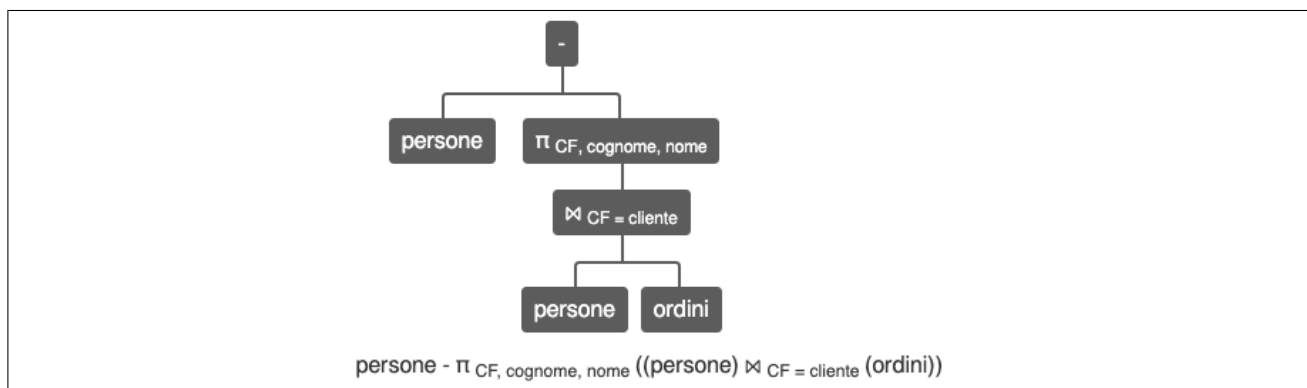
Domanda 1 (75%) Considerare il seguente schema relazionale

```
CREATE TABLE Persone ( CF text NOT NULL PRIMARY KEY,
                        cognome text,
                        nome text );
CREATE TABLE Ordini ( codice integer NOT NULL PRIMARY KEY,
                        cliente text NOT NULL REFERENCES persone,
                        descrizione text NOT NULL);
CREATE TABLE RateEmesse ( numero integer NOT NULL PRIMARY KEY,
                            ordine integer NOT NULL REFERENCES ordini ,
                            data date NOT NULL,
                            importo integer NOT NULL,
                            datapagamento date );
```

Si noti che ogni ordine è relativo ad un cliente e che ogni rata emessa è relativa ad un ordine.

Formulare la seguente interrogazione in algebra relazionale

1. Mostrare codice fiscale, nome e cognome delle persone che non hanno nessun ordine



Formulare le seguenti interrogazioni in SQL

2. Per ciascun ordine che abbia rate emesse, mostrare l'importo totale delle rate stesse

```
select ordine, sum(importo) as importoTotale
from rateemesse
group by ordine
```

3. Per ciascuna persona, mostrare il numero di ordini, mostrando anche (con il valore 0) le persone che non hanno ordini.

```
select cf as persona, count(codice) as numeroordini
from persone left join ordini on cf=cliente
group by cf
```

oppure

```
select cliente as persona, count(*) as numeroordini
from ordini
group by cliente
union
select cf as persona, 0
from persone
where cf not in (select cliente from ordini)
```

oppure

```
select cliente as persona, count(*) as numeroordini
from ordini
group by cliente
union
select cf as persona, 0
from persone join ordini on cf=cliente
where codice is null
```

4. Per ciascun ordine che abbia rate emesse, mostrare il totale delle rate emesse, il totale delle rate pagate e l'eventuale “debito”, cioè la differenza fra il totale delle rate emesse e il totale delle rate pagate. Per semplicità, per questa interrogazione, supporre che tutti gli ordini abbiano almeno una rata emessa e almeno una rata pagata.

```
create or replace view totalerateemesse as
select ordine, sum(importo) as importototale
from rateemesse
group by ordine;

create or replace view totaleratepagate as
select ordine, sum(importo) as importototalepagato
from rateemesse
where datapagamento is not null
group by ordine;

select e.ordine, importototale, importototalepagato,
       importototale-importototalepagato as debito
from totalerateemesse e join totaleratepagate p on e.ordine = p.ordine

oppure

select ordine, sum(importo) as importototale,
       sum(case when datapagamento is not null then importo end) as importototalepagato,
       sum(importo)-sum(case when datapagamento is not null then importo end) as debito
from rateemesse
group by ordine
```

5. Mostrare le informazioni sull'ordine per il quale è massimo il “debito” di cui all'interrogazione precedente

```
create or replace view debiti as
select e.ordine, importototale, importototalepagato,
       importototale-importototalepagato as debito
from totalerateemesse e join totaleratepagate p on e.ordine = p.ordine;

select ordini.*, debito
from ordini join debiti on codice=ordine
where debito = (select max(debito) from debiti)
```

Domanda 2 (15%) Considerare le seguenti relazioni (tutte senza valori nulli)

- $R_1(\underline{A}, B)$, con vincolo di integrità referenziale fra B e la chiave D di R_2 e con cardinalità $C_1 = 1000$
- $R_2(\underline{D}, E, F, G)$, con vincolo di integrità referenziale fra F, G e la chiave H, P di R_3 e con cardinalità $C_2 = 400$
- $R_3(\underline{H}, \underline{P}, Q)$, 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)
$\pi_{EF}(R_2)$	1	C_2	1	400
$(R_1 \bowtie_{(B=D)} R_2) \bowtie_{(F=H) \wedge (G=P)} R_3$	C_1	C_1	1000	1000
$R_2 \bowtie_{(F=H) \wedge (G=P)} R_3$	C_2	C_2	400	400
$R_1 \bowtie_{(B=P)} R_3$	0	$C_1 \times C_3$	0	500.000

Domanda 3 (10%)

Con riferimento ad una relazione **GIOCATORI**(Codice, Nome, Altezza, Ruolo), scrivere le interrogazioni SQL che calcolano, per ciascun ruolo, l'altezza media dei giocatori di tale 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 -1 per indicare che l'altezza non è nota

Soluzione

```
SELECT Ruolo, AVG(Altezza) AS AltezzaMedia
FROM Giocatori
WHERE Altezza <> -1
GROUP BY Ruolo
```

Basi di dati — 18 novembre 2019 — Prova parziale — Compito B

Possibili soluzioni

Tempo a disposizione: un'ora.

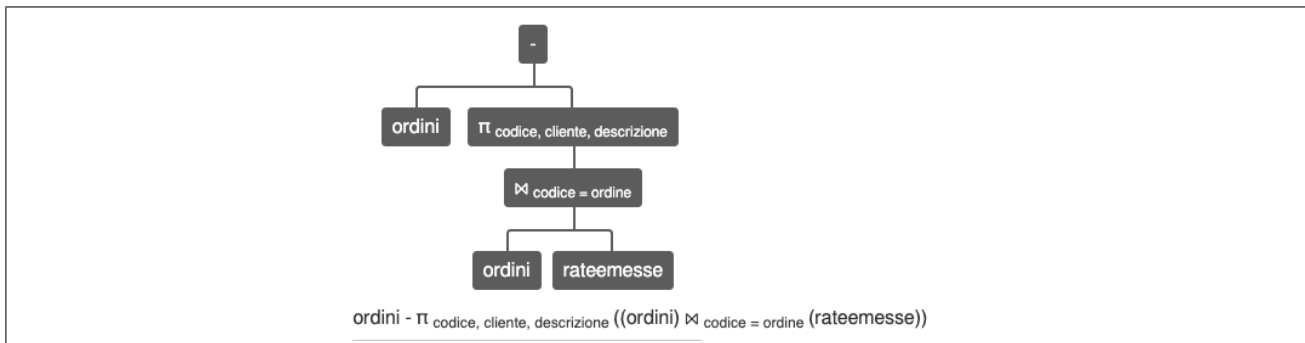
Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (75%) Considerare il seguente schema relazionale

```
CREATE TABLE Persone ( CF text NOT NULL PRIMARY KEY,
                        cognome text,
                        nome text );
CREATE TABLE Ordini ( codice integer NOT NULL PRIMARY KEY,
                        cliente text NOT NULL REFERENCES persone,
                        descrizione text NOT NULL);
CREATE TABLE RateEmesse ( numero integer NOT NULL PRIMARY KEY,
                            ordine integer NOT NULL REFERENCES ordini ,
                            data date NOT NULL,
                            importo integer NOT NULL,
                            datapagamento date );
```

Si noti che ogni ordine è relativo ad un cliente e che ogni rata emessa è relativa ad un ordine.
Formulare la seguente interrogazione in **algebra relazionale**

1. Mostrare codice, cliente e descrizione degli ordini per i quali non c'è nessuna rata emessa



Formulare le seguenti interrogazioni in **SQL**

2. Per ciascuna persona che abbia ordini, mostrare il numero totale di tali ordini

```
select cf, cognome, nome, count(*) as numeroOrdini
from persone join ordini on cf=cliente
group by cf, cognome, nome
```

3. Per ciascun ordine, mostrare l'importo totale delle rate emesse, mostrando anche (con il valore 0) gli ordini che non hanno rate emesse.

```
select ordine, sum(importo) as importototale
from rateemesse
group by ordine
union
select codice as ordine, 0
from ordini
where codice not in (select ordine from rateemesse)
```

oppure

```
select ordine, sum(importo) as importototale
from rateemesse
group by ordine
union
select codice as ordine, 0
from ordini left join rateemesse on codice=ordine
where importo is null
```

oppure

```
select codice, coalesce(sum(importo),0) as totale
from ordini left join rateemesse on codice=ordine
group by codice
```

4. Per ciascun ordine che abbia rate emesse, mostrare il totale delle rate emesse, il totale delle rate pagate e l'eventuale “debito”, cioè la differenza fra il totale delle rate emesse e il totale delle rate pagate. Per semplicità, per questa interrogazione, supporre che tutti gli ordini abbiano almeno una rata emessa e almeno una rata pagata.

```
create or replace view totalerateemesse as
select ordine, sum(importo) as importototale
from rateemesse
group by ordine;

create or replace view totaleratepagate as
select ordine, sum(importo) as importototalepagato
from rateemesse
where datapagamento is not null
group by ordine;

select e.ordine, importototale, importototalepagato,
       importototale-importototalepagato as debito
from totalerateemesse e join totaleratepagate p on e.ordine = p.ordine

oppure

select ordine, sum(importo) as importototale,
       sum(case when datapagamento is not null then importo end) as importototalepagato,
       sum(importo)-sum(case when datapagamento is not null then importo end) as debito
from rateemesse
group by ordine
```

5. Mostrare le informazioni sull'ordine per il quale è massimo il “debito” di cui all'interrogazione precedente

```
create or replace view debiti as
select e.ordine, importototale, importototalepagato,
       importototale-importototalepagato as debito
from totalerateemesse e join totaleratepagate p on e.ordine = p.ordine;

select ordini.*, debito
from ordini join debiti on codice=ordine
where debito = (select max(debito) from debiti)
```


Domanda 2 (15%) Considerare le seguenti relazioni (tutte senza valori nulli)

- $R_1(\underline{A}, B)$, con vincolo di integrità referenziale fra B e la chiave D di R_2 e con cardinalità $L_1 = 1000$
- $R_2(\underline{D}, E, F, G)$, con vincolo di integrità referenziale fra F, G e la chiave H, P di R_3 e con cardinalità $L_2 = 400$
- $R_3(\underline{H}, \underline{P}, Q)$, con cardinalità $L_3 = 500$

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)
$R_1 \bowtie_{(B=D)} R_2$	L_1	L_1	1000	1000
$(R_1 \bowtie_{(B=D)} R_2) \bowtie_{(F=H) \wedge (G=P)} R_3$	L_1	L_1	1000	1000
$\pi_{HP}(R_3)$	L_3	L_3	500	500
$R_3 \bowtie_{(Q=A)} R_1$	0	L_3	0	500

Domanda 3 (10%)

Con riferimento ad una relazione IMPIEGATI(Codice, Nome, Retribuzione, Livello), scrivere le interrogazioni SQL che calcolano, per ciascun livello, la retribuzione media degli impiegati di tale livello, nei due casi seguenti:

- si usa il valore nullo per indicare che la retribuzione non è nota

Soluzione

```
SELECT Livello, AVG(Retribuzione) AS RetribuzioneMedia
FROM Impiegati
GROUP BY Livello
```

- si usa il valore 0 per indicare che la retribuzione non è nota

Soluzione

```
SELECT Livello, AVG(Retribuzione) AS RetribuzioneMedia
FROM Impiegati
WHERE Retribuzione <> 0
GROUP BY Livello
```