

Basi di dati — 13 novembre 2017 — Prova parziale — Compito A
Tempo a disposizione: un'ora.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (25%) Considerare lo schema con le seguenti relazioni

```
create table studenti (matricola numeric not null primary key,  
                       cognome char(20) not null,  
                       nome char(20) not null,  
                       eta numeric);  
create table corsi (codice numeric not null primary key,  
                   titolo char(20) not null,  
                   CFU numeric not null);  
create table esami (corso numeric not null references corsi(codice),  
                   studente numeric not null references studenti(matricola),  
                   data date not null,  
                   voto numeric not null,  
                   primary key (corso, studente));
```

con le seguenti cardinalità

- **studenti**: cardinalità $S = 1000$
- **corsi**: cardinalità $C = 200$
- **esami**: cardinalità $E = 10.000$

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 matricola, codice FROM studenti, corsi</pre>		
<pre>SELECT * FROM studenti JOIN esami ON matricola = studente WHERE voto > 27</pre>		
<pre>SELECT matricola, codice FROM studenti, esami, corsi WHERE matricola = studente AND corso = codice AND voto > 24</pre>		

Basi di dati I — 13 novembre 2017 — Compito A

Domanda 2 (55%) Considerare nuovamente lo schema utilizzato nella domanda precedente

```
create table studenti (matricola numeric not null primary key,  
                      cognome char(20) not null,  
                      nome char(20) not null,  
                      eta numeric);  
create table corsi (codice numeric not null primary key,  
                   titolo char(20) not null,  
                   CFU numeric not null);  
create table esami (corso numeric not null references corsi(codice),  
                   studente numeric not null references studenti(matricola),  
                   data date not null,  
                   voto numeric not null,  
                   primary key (corso, studente));
```

Formulare la seguente interrogazione in algebra relazionale

1. Mostrare codice e titolo dei corsi per i quali non è stato registrato nessun esame.

Formulare le seguenti interrogazioni in SQL

2. Per ciascuno studente, mostrare matricola, numero di CFU conseguiti e media dei voti.

3. Per ciascun corso, mostrare codice, numero di esami registrati e media dei voti.

Basi di dati I — 13 novembre 2017 — Compito A

4. Per ciascun corso, mostrare codice e numero di esami registrati, includendo anche i corsi per i quali non è stato registrato alcun esame (con il valore 0 per il numero di esami).

5. Mostrare matricola, cognome e nome dello studente che ha conseguito il maggior numero di crediti.

Domanda 3 (20%)

Considerare una relazione

STIPENDI(Matricola,StipLordo,Tasse,Netto,Verifica)

e definire su di essa

1. un vincolo che imponga che, se il valore di *Verifica* è “OK”, allora *Netto* è uguale alla differenza fra *StipLordo* e *Tasse* (si noti che non si vuole invece imporre nessuna condizione se il valore di *Verifica* è diverso da “OK”).

2. un vincolo che imponga che il valore di *Verifica* è “OK” se e solo se *Netto* è uguale alla differenza fra *StipLordo* e *Tasse*.

Basi di dati — 13 novembre 2017 — Prova parziale — Compito B
Tempo a disposizione: un'ora.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (25%) Considerare lo schema con le seguenti relazioni

```
create table studenti (matricola numeric not null primary key,
                      cognome char(20) not null,
                      nome char(20) not null,
                      eta numeric);
create table corsi (codice numeric not null primary key,
                  titolo char(20) not null,
                  CFU numeric not null);
create table esami (corso numeric not null references corsi(codice),
                  studente numeric not null references studenti(matricola),
                  data date not null,
                  voto numeric not null,
                  primary key (corso, studente));
```

con le seguenti cardinalità

- **studenti**: cardinalità $S = 1000$
- **corsi**: cardinalità $C = 200$
- **esami**: cardinalità $E = 10.000$

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 matricola, codice FROM studenti, corsi WHERE eta > 20</pre>		
<pre>SELECT * FROM studenti JOIN esami ON matricola = studente WHERE voto > 27</pre>		
<pre>SELECT matricola, codice FROM studenti, esami, corsi WHERE matricola = studente AND corso = codice</pre>		

Basi di dati I — 13 novembre 2017 — Compito B

Domanda 2 (55%) Considerare nuovamente lo schema utilizzato nella domanda precedente

```
create table studenti (matricola numeric not null primary key,  
                      cognome char(20) not null,  
                      nome char(20) not null,  
                      eta numeric);  
create table corsi (codice numeric not null primary key,  
                   titolo char(20) not null,  
                   CFU numeric not null);  
create table esami (corso numeric not null references corsi(codice),  
                   studente numeric not null references studenti(matricola),  
                   data date not null,  
                   voto numeric not null,  
                   primary key (corso, studente));
```

Formulare la seguente interrogazione in algebra relazionale

1. Mostrare matricola e cognome degli studenti per i quali non è stato registrato nessun esame

Formulare le seguenti interrogazioni in SQL

2. Per ciascun corso, mostrare codice, numero di esami registrati e media dei voti.

3. Per ciascuno studente, mostrare matricola, numero di CFU conseguiti e media dei voti.

Basi di dati I — 13 novembre 2017 — Compito B

4. Per ciascuno studente, mostrare matricola e numero di CFU conseguiti, includendo anche gli studenti che non hanno superato esami (con valore 0 per il numero di CFU conseguiti).

5. Mostrare codice e titolo del corso per il quale sono stati registrati più esami.

Domanda 3 (20%)

Considerare una relazione

PAGHE(Matricola,StipLordo,Ritenute,StipNetto,OK)

e definire su di essa

1. un vincolo che imponga che, se il valore di OK è "OK", allora StipNetto è uguale alla differenza fra StipLordo e Ritenute (si noti che non si vuole invece imporre nessuna condizione se il valore di OK è diverso da "OK").

2. un vincolo che imponga che il valore di OK è "OK" se e solo se StipNetto è uguale alla differenza fra StipLordo e Ritenute.

Basi di dati — 13 novembre 2017 — Prova parziale — Compito C
Tempo a disposizione: un'ora.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (25%) Considerare lo schema con le seguenti relazioni

```
create table studenti (matricola numeric not null primary key,  
                      cognome char(20) not null,  
                      nome char(20) not null,  
                      eta numeric);  
create table corsi (codice numeric not null primary key,  
                   titolo char(20) not null,  
                   CFU numeric not null);  
create table esami (corso numeric not null references corsi(codice),  
                   studente numeric not null references studenti(matricola),  
                   data date not null,  
                   voto numeric not null,  
                   primary key (corso, studente));
```

con le seguenti cardinalità

- **studenti**: cardinalità $S = 1000$
- **corsi**: cardinalità $C = 200$
- **esami**: cardinalità $E = 10.000$

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 matricola, codice FROM studenti, corsi</pre>		
<pre>SELECT * FROM studenti JOIN esami ON matricola = studente</pre>		
<pre>SELECT matricola, codice FROM studenti, esami, corsi WHERE matricola = studente AND corso = codice AND voto > 24</pre>		

Basi di dati I — 13 novembre 2017 — Compito C

Domanda 2 (55%) Considerare nuovamente lo schema utilizzato nella domanda precedente

```
create table studenti (matricola numeric not null primary key,  
                      cognome char(20) not null,  
                      nome char(20) not null,  
                      eta numeric);  
create table corsi (codice numeric not null primary key,  
                  titolo char(20) not null,  
                  CFU numeric not null);  
create table esami (corso numeric not null references corsi(codice),  
                  studente numeric not null references studenti(matricola),  
                  data date not null,  
                  voto numeric not null,  
                  primary key (corso, studente));
```

Formulare la seguente interrogazione in algebra relazionale

1. Mostrare codice e titolo dei corsi per i quali non è stato registrato nessun esame.

Formulare le seguenti interrogazioni in SQL

2. Per ciascuno studente, mostrare matricola, numero di CFU conseguiti e media dei voti.

3. Per ciascun corso, mostrare codice, numero di esami registrati e media dei voti.

Basi di dati I — 13 novembre 2017 — Compito C

4. Per ciascun corso, mostrare codice e numero di esami registrati, includendo anche i corsi per i quali non è stato registrato alcun esame (con il valore 0 per il numero di esami).

5. Mostrare codice e titolo del corso per il quale sono stati registrati più esami.

Domanda 3 (20%)

Considerare una relazione

RETRIBUZIONI(Matricola,Lordo,Imposte,StipNetto,Verifica)

e definire su di essa

1. un vincolo che imponga che, se il valore di *Verifica* è “OK”, allora *StipNetto* è uguale alla differenza fra *Lordo* e *Imposte* (si noti che non si vuole invece imporre nessuna condizione se il valore di *Verifica* è diverso da “OK”).

2. un vincolo che imponga che il valore di *Verifica* è “OK” se e solo se *StipNetto* è uguale alla differenza fra *Lordo* e *Imposte*.

Basi di dati — 13 novembre 2017 — Prova parziale — Compito D
Tempo a disposizione: un'ora.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (25%) Considerare lo schema con le seguenti relazioni

```
create table studenti (matricola numeric not null primary key,  
                      cognome char(20) not null,  
                      nome char(20) not null,  
                      eta numeric);  
create table corsi (codice numeric not null primary key,  
                   titolo char(20) not null,  
                   CFU numeric not null);  
create table esami (corso numeric not null references corsi(codice),  
                   studente numeric not null references studenti(matricola),  
                   data date not null,  
                   voto numeric not null,  
                   primary key (corso, studente));
```

con le seguenti cardinalità

- **studenti**: cardinalità $S = 1000$
- **corsi**: cardinalità $C = 200$
- **esami**: cardinalità $E = 10.000$

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 matricola, codice FROM studenti, corsi WHERE eta > 20</pre>		
<pre>SELECT * FROM studenti JOIN esami ON matricola = studente</pre>		
<pre>SELECT matricola, codice FROM studenti, esami, corsi WHERE matricola = studente AND corso = codice</pre>		

Basi di dati I — 13 novembre 2017 — Compito D

Domanda 2 (55%) Considerare nuovamente lo schema utilizzato nella domanda precedente

```
create table studenti (matricola numeric not null primary key,  
                      cognome char(20) not null,  
                      nome char(20) not null,  
                      eta numeric);  
create table corsi (codice numeric not null primary key,  
                   titolo char(20) not null,  
                   CFU numeric not null);  
create table esami (corso numeric not null references corsi(codice),  
                   studente numeric not null references studenti(matricola),  
                   data date not null,  
                   voto numeric not null,  
                   primary key (corso, studente));
```

Formulare la seguente interrogazione in algebra relazionale

1. Mostrare matricola e cognome degli studenti per i quali non è stato registrato nessun esame

Formulare le seguenti interrogazioni in SQL

2. Per ciascun corso, mostrare codice, numero di esami registrati e media dei voti.

3. Per ciascuno studente, mostrare matricola, numero di CFU conseguiti e media dei voti.

Basi di dati I — 13 novembre 2017 — Compito D

4. Per ciascuno studente, mostrare matricola e numero di CFU conseguiti, includendo anche gli studenti che non hanno superato esami (con valore 0 per il numero di CFU conseguiti).

5. Mostrare matricola, cognome e nome dello studente che ha conseguito il maggior numero di crediti.

Domanda 3 (20%)

Considerare una relazione

SALARI(Matricola,StipLordo,Trattenute,Netto,OK)

e definire su di essa

1. un vincolo che imponga che, se il valore di OK è "OK", allora Netto è uguale alla differenza fra StipLordo e Trattenute (si noti che non si vuole invece imporre nessuna condizione se il valore di OK è diverso da "OK").

2. un vincolo che imponga che il valore di OK è "OK" se e solo se Netto è uguale alla differenza fra StipLordo e Trattenute.