

Basi di dati — 15 novembre 2013 — Prova parziale — Compito A
Tempo a disposizione: un'ora. Libri chiusi.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (15%) Si supponga di voler rappresentare in una base di dati relazionale le informazioni relative alla programmazione di una giornata in un cinema multisala, che vengono pubblicate con avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Film	Psycho	Shining	Sabrina
Spettacoli	20:00	21:10	20:20
	22:00		22:20

Mostrare gli schemi delle relazioni da utilizzare (con attributi e vincoli di chiave e di integrità referenziale) e l'istanza corrispondente ai dati sopra mostrati.

Domanda 2 (15%) Ripetere quanto sopra con riferimento ad avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Spettacoli	Psycho 20:00	Shining 21:10	Sabrina 20:20
	Casanova 22:00		Sabrina 22:20

Domanda 3 (20%) Considerare la base di dati relazionale definita per mezzo delle seguenti istruzioni:

```
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));
```

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità $S = 10.000$ (studenti), $C = 1.000$ (corsi) e $E = 40.000$ (esami), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT matricola, codice FROM studenti, corsi				
SELECT * FROM studenti, esami WHERE matricola = studente				
SELECT matricola, codice FROM studenti, esami, corsi WHERE matricola = studente AND corso = codice				

Domanda 4 (20%) Con riferimento alla base di dati usata nella domanda precedente formulare le seguenti interrogazioni in algebra relazionale:

1. trovare matricole e cognomi degli studenti che hanno preso almeno un trenta

2. trovare le matricole degli studenti che hanno sostenuto almeno due esami

Domanda 5 (30%) Con riferimento alla base di dati usata nelle domande precedenti, formulare le seguenti interrogazioni in SQL

1. trovare codici e titoli di corsi nei cui esami è stato assegnato almeno un trenta

2. trovare le coppie di studenti (mostrare le sole matricole) per i quali uno dei due ha riportato un voto più alto in tutti gli esami superati da entrambi.

3. trovare lo studente con la media più alta; mostrare i dati dello studente, la media in questione e il numero di esami superati

Basi di dati — 15 novembre 2013 — Prova parziale — Compito B
Tempo a disposizione: un'ora. Libri chiusi.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (15%) Si supponga di voler rappresentare in una base di dati relazionale le informazioni relative alla programmazione di una giornata in un cinema multisala, che vengono pubblicate con avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Film	Frenzy	L'appartamento	Ludwig
Spettacoli	20:00	21:10	20:20
	22:00		22:20

Mostrare gli schemi delle relazioni da utilizzare (con attributi e vincoli di chiave e di integrità referenziale) e l'istanza corrispondente ai dati sopra mostrati.

Domanda 2 (15%) Ripetere quanto sopra con riferimento ad avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Spettacoli	Frenzy 20:00	L'appartamento 21:10	Ludwig 20:20
	Casanova 22:00		Ludwig 22:20

Domanda 3 (20%) Considerare la base di dati relazionale definita per mezzo delle seguenti istruzioni:

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

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità $S = 10.000$ (studenti), $M = 1.000$ (materie) e $E = 40.000$ (esami), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT matricola, codice FROM studenti, materie				
SELECT * FROM studenti, esami WHERE matricola = studente				
SELECT matricola, codice FROM studenti, esami, materie WHERE matricola = studente AND materia = codice				

Domanda 4 (20%) Con riferimento alla base di dati usata nella domanda precedente formulare le seguenti interrogazioni in algebra relazionale:

1. trovare codici e titoli di materie nei cui esami è stato assegnato almeno un trenta

2. trovare le matricole degli studenti che hanno sostenuto almeno due esami

Domanda 5 (30%) Con riferimento alla base di dati usata nelle domande precedenti, formulare le seguenti interrogazioni in SQL

1. trovare matricole e cognomi degli studenti che hanno preso almeno un trenta

2. trovare le coppie di studenti (mostrare le sole matricole) per i quali uno dei due ha riportato un voto più alto in tutti gli esami superati da entrambi.

3. trovare lo studente con la media più alta; mostrare i dati dello studente, la media in questione e il numero di esami superati

Basi di dati — 15 novembre 2013 — Prova parziale — Compito C
Tempo a disposizione: un'ora. Libri chiusi.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (15%) Si supponga di voler rappresentare in una base di dati relazionale le informazioni relative alla programmazione di una giornata in un cinema multisala, che vengono pubblicate con avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Film	Accattone	Otto e mezzo	Ossessione
Spettacoli	20:00	21:10	20:20
	22:00		22:20

Mostrare gli schemi delle relazioni da utilizzare (con attributi e vincoli di chiave e di integrità referenziale) e l'istanza corrispondente ai dati sopra mostrati.

Domanda 2 (15%) Ripetere quanto sopra con riferimento ad avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Spettacoli	Accattone 20:00	Otto e mezzo 21:10	Ossessione 20:20
	Casanova 22:00		Ossessione 22:20

Domanda 3 (20%) Considerare la base di dati relazionale definita per mezzo delle seguenti istruzioni:

```
create table studenti (matricola numeric not null primary key,
                      cognome char(20) not null,
                      nome char(20) not null,
                      etÃ  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));
```

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità $S = 10.000$ (studenti), $C = 1.000$ (corsi) e $E = 40.000$ (esami), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT matricola, codice FROM studenti, corsi				
SELECT * FROM studenti, esami WHERE matricola = studente				
SELECT matricola, codice FROM studenti, esami, corsi WHERE matricola = studente AND corso = codice				

Domanda 4 (20%) Con riferimento alla base di dati usata nella domanda precedente formulare le seguenti interrogazioni in algebra relazionale:

1. trovare codici e titoli di corsi nei cui esami è stato assegnato almeno un trenta

2. trovare le matricole degli studenti che hanno sostenuto almeno due esami

Domanda 5 (30%) Con riferimento alla base di dati usata nelle domande precedenti, formulare le seguenti interrogazioni in SQL

1. trovare matricole e cognomi degli studenti che hanno preso almeno un trenta

2. trovare le coppie di studenti (mostrare le sole matricole) per i quali uno dei due ha riportato un voto più alto in tutti gli esami superati da entrambi.

3. trovare lo studente con la media più alta; mostrare i dati dello studente, la media in questione e il numero di esami superati

Basi di dati — 15 novembre 2013 — Prova parziale — Compito D
Tempo a disposizione: un'ora. Libri chiusi.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (15%) Si supponga di voler rappresentare in una base di dati relazionale le informazioni relative alla programmazione di una giornata in un cinema multisala, che vengono pubblicate con avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Film	Senso	Paisà	Roma città aperta
Spettacoli	20:00	21:10	20:20
	22:00		22:20

Mostrare gli schemi delle relazioni da utilizzare (con attributi e vincoli di chiave e di integrità referenziale) e l'istanza corrispondente ai dati sopra mostrati.

Domanda 2 (15%) Ripetere quanto sopra con riferimento ad avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Spettacoli	Senso 20:00	Paisà 21:10	Roma città aperta 20:20
	Casanova 22:00		Roma città aperta 22:20

Domanda 3 (20%) Considerare la base di dati relazionale definita per mezzo delle seguenti istruzioni:

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

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità $S = 10.000$ (studenti), $M = 1.000$ (materie) e $E = 40.000$ (esami), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT matricola, codice FROM studenti, materie				
SELECT * FROM studenti, esami WHERE matricola = studente				
SELECT matricola, codice FROM studenti, esami, materie WHERE matricola = studente AND materia = codice				

Domanda 4 (20%) Con riferimento alla base di dati usata nella domanda precedente formulare le seguenti interrogazioni in algebra relazionale:

1. trovare matricole e cognomi degli studenti che hanno preso almeno un trenta

2. trovare le matricole degli studenti che hanno sostenuto almeno due esami

Domanda 5 (30%) Con riferimento alla base di dati usata nelle domande precedenti, formulare le seguenti interrogazioni in SQL

1. trovare codici e titoli di materie nei cui esami è stato assegnato almeno un trenta

2. trovare le coppie di studenti (mostrare le sole matricole) per i quali uno dei due ha riportato un voto più alto in tutti gli esami superati da entrambi.

3. trovare lo studente con la media più alta; mostrare i dati dello studente, la media in questione e il numero di esami superati

Possibili soluzioni

Tempo a disposizione: un'ora. Libri chiusi.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (15%) Si supponga di voler rappresentare in una base di dati relazionale le informazioni relative alla programmazione di una giornata in un cinema multisala, che vengono pubblicate con avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Film	Psycho	Shining	Sabrina
Spettacoli	20:00	21:10	20:20
	22:00		22:20

Soluzione

Sale		
Codice	Posti	Film
1	150	Psycho
2	100	Shining
3	200	Sabrina

Spettacoli	
Sala	Orario
1	20:00
1	22:00
2	21:10
3	20:20
3	22:20

oppure

Spettacoli		
Sala	Spettacolo	Orario
1	1	20:00
1	2	22:00
2	1	21:10
3	1	20:20
3	2	22:20

Vincolo di integrità referenziale fra Sala di Spettacoli e la chiave primaria di Sale

Mostrare gli schemi delle relazioni da utilizzare (con attributi e vincoli di chiave e di integrità referenziale) e l'istanza corrispondente ai dati sopra mostrati.

Domanda 2 (15%) Ripetere quanto sopra con riferimento ad avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Spettacoli	Psycho 20:00	Shining 21:10	Sabrina 20:20
	Casanova 22:00		Sabrina 22:20

Soluzione

Sale	
Codice	Posti
1	150
2	100
3	200

Spettacoli		
Sala	Orario	Film
1	20:00	Psycho
1	22:00	Casanova
2	21:10	Shining
3	20:20	Sabrina
3	22:20	Sabrina

oppure

Spettacoli			
Sala	Spettacolo	Orario	Film
1	1	20:00	Psycho
1	2	22:00	Casanova
2	1	21:10	Shining
3	1	20:20	Sabrina
3	2	22:20	Sabrina

Vincolo di integrità referenziale fra Sala di Spettacoli e la chiave primaria di Sale

Domanda 3 (20%) Considerare la base di dati relazionale definita per mezzo delle seguenti istruzioni:

```
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));
```

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità $S = 10.000$ (studenti), $C = 1.000$ (corsi) e $E = 40.000$ (esami), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT matricola, codice FROM studenti, corsi	$S \times C$	$S \times C$	10.000.000	10.000.000
SELECT * FROM studenti, esami WHERE matricola = studente	E	E	40.000	40.000
SELECT matricola, codice FROM studenti, esami, corsi WHERE matricola = studente AND corso = codice	E	E	40.000	40.000

Domanda 4 (20%) Con riferimento alla base di dati usata nella domanda precedente formulare le seguenti interrogazioni in algebra relazionale:

1. trovare matricole e cognomi degli studenti che hanno preso almeno un trenta

$$\pi_{matricola, cognome}(studenti \bowtie_{matricola=studente} \sigma_{voto=30}(esami))$$

2. trovare le matricole degli studenti che hanno sostenuto almeno due esami

$$\pi_{studente}(\sigma_{corso \neq corso'}(esami \bowtie_{studente=studente'} \rho_{X \leftarrow X'}(esami)))$$

Domanda 5 (30%) Con riferimento alla base di dati usata nelle domande precedenti, formulare le seguenti interrogazioni in SQL

1. trovare codici e titoli di corsi nei cui esami è stato assegnato almeno un trenta

```
SELECT DISTINCT codice, titolo
FROM corsi JOIN esami ON codice = corso
WHERE voto = 30
```

2. trovare le coppie di studenti (mostrare le sole matricole) per i quali uno dei due ha riportato un voto più alto in tutti gli esami superati da entrambi.

```
SELECT e1.studente, e2.studente
FROM esami e1, esami e2
WHERE e1.voto > e2.voto
AND e1.studente <> e2.studente
AND e1.corso = e2.corso
AND NOT EXISTS (
    SELECT *
    FROM esami e3, esami e4
    WHERE e3.corso = e4.corso
    AND e3.studente = e1.studente
    AND e4.studente = e2.studente
    AND e3.voto <= e4.voto )
```

3. trovare lo studente con la media più alta; mostrare i dati dello studente, la media in questione e il numero di esami superati

```
CREATE VIEW MediaVoti AS SELECT studente, AVG(voto) AS media, COUNT(*) AS numEsami
FROM esami
GROUP BY studente
```

```
SELECT studente, media, numEsami
FROM MediaVoti, studenti
WHERE studente = matricola
AND media = (SELECT MAX(media)
             FROM MediaVoti)
```

oppure

```
SELECT matricola, cognome, nome, AVG(voto), COUNT(*)
FROM esami join studenti on studente = matricola
GROUP BY matricola, cognome, nome
HAVING AVG(voto) >= ALL
    (SELECT AVG(voto)
     FROM esami
     GROUP BY studente)
```

Possibili soluzioni

Tempo a disposizione: un'ora. Libri chiusi.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (15%) Si supponga di voler rappresentare in una base di dati relazionale le informazioni relative alla programmazione di una giornata in un cinema multisala, che vengono pubblicate con avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Film	Frenzy	L'appartamento	Ludwig
Spettacoli	20:00 22:00	21:10	20:20 22:20

Soluzione

Sale		
Codice	Posti	Film
1	150	Frenzy
2	100	L'appartamento
3	200	Ludwig

Spettacoli	
Sala	Orario
1	20:00
1	22:00
2	21:10
3	20:20
3	22:20

oppure

Spettacoli		
Sala	Spettacolo	Orario
1	1	20:00
1	2	22:00
2	1	21:10
3	1	20:20
3	2	22:20

Vincolo di integrità referenziale fra Sala di Spettacoli e la chiave primaria di Sale

Mostrare gli schemi delle relazioni da utilizzare (con attributi e vincoli di chiave e di integrità referenziale) e l'istanza corrispondente ai dati sopra mostrati.

Domanda 2 (15%) Ripetere quanto sopra con riferimento ad avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Spettacoli	Frenzy 20:00 Casanova 22:00	L'appartamento 21:10	Ludwig 20:20 Ludwig 22:20

Soluzione

Sale	
Codice	Posti
1	150
2	100
3	200

Spettacoli		
Sala	Orario	Film
1	20:00	Frenzy
1	22:00	Casanova
2	21:10	L'appartamento
3	20:20	Ludwig
3	22:20	Ludwig

oppure

Spettacoli			
Sala	Spettacolo	Orario	Film
1	1	20:00	Frenzy
1	2	22:00	Casanova
2	1	21:10	L'appartamento
3	1	20:20	Ludwig
3	2	22:20	Ludwig

Vincolo di integrità referenziale fra Sala di Spettacoli e la chiave primaria di Sale

Domanda 3 (20%) Considerare la base di dati relazionale definita per mezzo delle seguenti istruzioni:

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

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità $S = 10.000$ (studenti), $M = 1.000$ (materie) e $E = 40.000$ (esami), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT matricola, codice FROM studenti, materie	$S \times M$	$S \times M$	10.000.000	10.000.000
SELECT * FROM studenti, esami WHERE matricola = studente	E	E	40.000	40.000
SELECT matricola, codice FROM studenti, esami, materie WHERE matricola = studente AND materia = codice	E	E	40.000	40.000

Domanda 4 (20%) Con riferimento alla base di dati usata nella domanda precedente formulare le seguenti interrogazioni in algebra relazionale:

- trovare codici e titoli di materie nei cui esami è stato assegnato almeno un trenta

$$\pi_{\text{codice, titolo}}(\text{materie} \bowtie_{\text{codice=materia}} \sigma_{\text{voto} \geq 30}(\text{esami}))$$

- trovare le matricole degli studenti che hanno sostenuto almeno due esami

$$\pi_{\text{studente}}(\sigma_{\text{materia} \neq \text{materia}'}(\text{esami} \bowtie_{\text{studente=studente}'} \rho_{X \leftarrow X'}(\text{esami})))$$

Domanda 5 (30%) Con riferimento alla base di dati usata nelle domande precedenti, formulare le seguenti interrogazioni in SQL

1. trovare matricole e cognomi degli studenti che hanno preso almeno un trenta

```
SELECT matricola, cognome
FROM studenti JOIN esami ON matricola = studente
WHERE voto = 30
```

2. trovare le coppie di studenti (mostrare le sole matricole) per i quali uno dei due ha riportato un voto più alto in tutti gli esami superati da entrambi.

```
SELECT e1.studente, e2.studente
FROM esami e1, esami e2
WHERE e1.voto > e2.voto
AND e1.studente <> e2.studente
AND e1.materia = e2.materia
AND NOT EXISTS (
    SELECT *
    FROM esami e3, esami e4
    WHERE e3.materia = e4.materia
    AND e3.studente = e1.studente
    AND e4.studente = e2.studente
    AND e3.voto <= e4.voto )
```

3. trovare lo studente con la media più alta; mostrare i dati dello studente, la media in questione e il numero di esami superati

```
CREATE VIEW MediaVoti AS SELECT studente, AVG(voto) AS media, COUNT(*) AS numEsami
FROM esami
GROUP BY studente
```

```
SELECT studente, media, numEsami
FROM MediaVoti, studenti
WHERE studente = matricola
AND media = (SELECT MAX(media)
             FROM MediaVoti)
```

oppure

```
SELECT matricola, cognome, nome, AVG(voto), COUNT(*)
FROM esami join studenti on studente = matricola
GROUP BY matricola, cognome, nome
HAVING AVG(voto) >= ALL
    (SELECT AVG(voto)
     FROM esami
     GROUP BY studente)
```

Possibili soluzioni
Tempo a disposizione: un'ora. Libri chiusi.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (15%) Si supponga di voler rappresentare in una base di dati relazionale le informazioni relative alla programmazione di una giornata in un cinema multisala, che vengono pubblicate con avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Film	Accattone	Otto e mezzo	Ossessione
Spettacoli	20:00	21:10	20:20
	22:00		22:20

Soluzione

Sale			Spettacoli		Spettacoli		
Codice	Posti	Film	Sala	Orario	Sala	Spettacolo	Orario
1	150	Accattone	1	20:00	1	1	20:00
2	100	Otto e mezzo	1	22:00	1	2	22:00
3	200	Ossessione	2	21:10	2	1	21:10
			3	20:20	3	1	20:20
			3	22:20	3	2	22:20

Vincolo di integrità referenziale fra Sala di Spettacoli e la chiave primaria di Sale

Mostrare gli schemi delle relazioni da utilizzare (con attributi e vincoli di chiave e di integrità referenziale) e l'istanza corrispondente ai dati sopra mostrati.

Domanda 2 (15%) Ripetere quanto sopra con riferimento ad avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Spettacoli	Accattone 20:00	Otto e mezzo 21:10	Ossessione 20:20
	Casanova 22:00		Ossessione 22:20

Soluzione

Sale		Spettacoli			Spettacoli			
Codice	Posti	Sala	Orario	Film	Sala	Spettacolo	Orario	Film
1	150	1	20:00	Accattone	1	1	20:00	Accattone
2	100	1	22:00	Casanova	1	2	22:00	Casanova
3	200	2	21:10	Otto e mezzo	2	1	21:10	Otto e mezzo
		3	20:20	Ossessione	3	1	20:20	Ossessione
		3	22:20	Ossessione	3	2	22:20	Ossessione

Vincolo di integrità referenziale fra Sala di Spettacoli e la chiave primaria di Sale

Domanda 3 (20%) Considerare la base di dati relazionale definita per mezzo delle seguenti istruzioni:

```
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));
```

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità $S = 10.000$ (studenti), $C = 1.000$ (corsi) e $E = 40.000$ (esami), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT matricola, codice FROM studenti, corsi	$S \times C$	$S \times C$	10.000.000	10.000.000
SELECT * FROM studenti, esami WHERE matricola = studente	E	E	40.000	40.000
SELECT matricola, codice FROM studenti, esami, corsi WHERE matricola = studente AND corso = codice	E	E	40.000	40.000

Domanda 4 (20%) Con riferimento alla base di dati usata nella domanda precedente formulare le seguenti interrogazioni in algebra relazionale:

- trovare codici e titoli di corsi nei cui esami è stato assegnato almeno un trenta

$$\pi_{\text{codice, titolo}}(\text{corsi} \bowtie_{\text{codice}=\text{corso}} \sigma_{\text{voto} \geq 30}(\text{esami}))$$

- trovare le matricole degli studenti che hanno sostenuto almeno due esami

$$\pi_{\text{studente}}(\sigma_{\text{corso} \neq \text{corso}'}(\text{esami} \bowtie_{\text{studente}=\text{studente}'} \rho_{X \leftarrow X'}(\text{esami})))$$

Domanda 5 (30%) Con riferimento alla base di dati usata nelle domande precedenti, formulare le seguenti interrogazioni in SQL

1. trovare matricole e cognomi degli studenti che hanno preso almeno un trenta

```
SELECT matricola, cognome
FROM studenti JOIN esami ON matricola = studente
WHERE voto = 30
```

2. trovare le coppie di studenti (mostrare le sole matricole) per i quali uno dei due ha riportato un voto più alto in tutti gli esami superati da entrambi.

```
SELECT e1.studente, e2.studente
FROM esami e1, esami e2
WHERE e1.voto > e2.voto
AND e1.studente <> e2.studente
AND e1.corso = e2.corso
AND NOT EXISTS (
    SELECT *
    FROM esami e3, esami e4
    WHERE e3.corso = e4.corso
    AND e3.studente = e1.studente
    AND e4.studente = e2.studente
    AND e3.voto <= e4.voto )
```

3. trovare lo studente con la media più alta; mostrare i dati dello studente, la media in questione e il numero di esami superati

```
CREATE VIEW MediaVoti AS SELECT studente, AVG(voto) AS media, COUNT(*) AS numEsami
FROM esami
GROUP BY studente
```

```
SELECT studente, media, numEsami
FROM MediaVoti, studenti
WHERE studente = matricola
AND media = (SELECT MAX(media)
             FROM MediaVoti)
```

oppure

```
SELECT matricola, cognome, nome, AVG(voto), COUNT(*)
FROM esami join studenti on studente = matricola
GROUP BY matricola, cognome, nome
HAVING AVG(voto) >= ALL
    (SELECT AVG(voto)
     FROM esami
     GROUP BY studente)
```

Possibili soluzioni

Tempo a disposizione: un'ora. Libri chiusi.

Cognome: _____ Nome: _____ Matricola: _____

Domanda 1 (15%) Si supponga di voler rappresentare in una base di dati relazionale le informazioni relative alla programmazione di una giornata in un cinema multisala, che vengono pubblicate con avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Film	Senso	Paisà	Roma città aperta
Spettacoli	20:00 22:00	21:10	20:20 22:20

Soluzione

Sale		
Codice	Posti	Film
1	150	Senso
2	100	Paisà
3	200	Roma città aperta

Spettacoli	
Sala	Orario
1	20:00
1	22:00
2	21:10
3	20:20
3	22:20

oppure

Spettacoli		
Sala	Spettacolo	Orario
1	1	20:00
1	2	22:00
2	1	21:10
3	1	20:20
3	2	22:20

Vincolo di integrità referenziale fra Sala di Spettacoli e la chiave primaria di Sale

Mostrare gli schemi delle relazioni da utilizzare (con attributi e vincoli di chiave e di integrità referenziale) e l'istanza corrispondente ai dati sopra mostrati.

Domanda 2 (15%) Ripetere quanto sopra con riferimento ad avvisi con la seguente struttura:

Sala	1	2	3
Posti	150	100	200
Spettacoli	Senso 20:00 Casanova 22:00	Paisà 21:10	Roma città aperta 20:20 Roma città aperta 22:20

Soluzione

Sale	
Codice	Posti
1	150
2	100
3	200

Spettacoli		
Sala	Orario	Film
1	20:00	Senso
1	22:00	Casanova
2	21:10	Paisà
3	20:20	Roma città aperta
3	22:20	Roma città aperta

oppure

Spettacoli			
Sala	Spettacolo	Orario	Film
1	1	20:00	Senso
1	2	22:00	Casanova
2	1	21:10	Paisà
3	1	20:20	Roma città aperta
3	2	22:20	Roma città aperta

Vincolo di integrità referenziale fra Sala di Spettacoli e la chiave primaria di Sale

Domanda 3 (20%) Considerare la base di dati relazionale definita per mezzo delle seguenti istruzioni:

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

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità $S = 10.000$ (studenti), $M = 1.000$ (materie) e $E = 40.000$ (esami), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT matricola, codice FROM studenti, materie	$S \times M$	$S \times M$	10.000.000	10.000.000
SELECT * FROM studenti, esami WHERE matricola = studente	E	E	40.000	40.000
SELECT matricola, codice FROM studenti, esami, materie WHERE matricola = studente AND materia = codice	E	E	40.000	40.000

Domanda 4 (20%) Con riferimento alla base di dati usata nella domanda precedente formulare le seguenti interrogazioni in algebra relazionale:

1. trovare matricole e cognomi degli studenti che hanno preso almeno un trenta

$$\pi_{matricola, cognome}(studenti \bowtie_{matricola=studente} \sigma_{voto=30}(esami))$$

2. trovare le matricole degli studenti che hanno sostenuto almeno due esami

$$\pi_{studente}(\sigma_{materia \neq materia'}(esami \bowtie_{studente=studente'} \rho_{X \leftarrow X'}(esami)))$$

Domanda 5 (30%) Con riferimento alla base di dati usata nelle domande precedenti, formulare le seguenti interrogazioni in SQL

1. trovare codici e titoli di materie nei cui esami è stato assegnato almeno un trenta

```
SELECT DISTINCT codice, titolo
FROM materie JOIN esami ON codice = materia
WHERE voto = 30
```

2. trovare le coppie di studenti (mostrare le sole matricole) per i quali uno dei due ha riportato un voto più alto in tutti gli esami superati da entrambi.

```
SELECT e1.studente, e2.studente
FROM esami e1, esami e2
WHERE e1.voto > e2.voto
AND e1.studente <> e2.studente
AND e1.materia = e2.materia
AND NOT EXISTS (
    SELECT *
    FROM esami e3, esami e4
    WHERE e3.materia = e4.materia
    AND e3.studente = e1.studente
    AND e4.studente = e2.studente
    AND e3.voto <= e4.voto )
```

3. trovare lo studente con la media più alta; mostrare i dati dello studente, la media in questione e il numero di esami superati

```
CREATE VIEW MediaVoti AS SELECT studente, AVG(voto) AS media, COUNT(*) AS numEsami
FROM esami
GROUP BY studente
```

```
SELECT studente, media, numEsami
FROM MediaVoti, studenti
WHERE studente = matricola
AND media = (SELECT MAX(media)
             FROM MediaVoti)
```

oppure

```
SELECT matricola, cognome, nome, AVG(voto), COUNT(*)
FROM esami join studenti on studente = matricola
GROUP BY matricola, cognome, nome
HAVING AVG(voto) >= ALL
    (SELECT AVG(voto)
     FROM esami
     GROUP BY studente)
```