

Informatica documentale
Laurea in Scienze della Comunicazione
(fax-simile di un tipico tema d'esame)

Cognome e nome: _____ Matricola: _____

Parte prima	Domanda 1	Domanda 2	Domanda 3	Totale
.../3	.../10	.../10	.../10	.../33

Istruzioni:

- È vietato portare all'esame libri, eserciziari, appunti e dispense. Chiunque venga trovato in possesso di documentazione relativa al corso – anche se non attinente alle domande proposte – vedrà annullata la propria prova.
- Scrivere solo sui fogli distribuiti, cancellando le parti di brutta con un tratto di penna. Non separare questi fogli.
- Tempo a disposizione: 1 ora e 30 minuti.

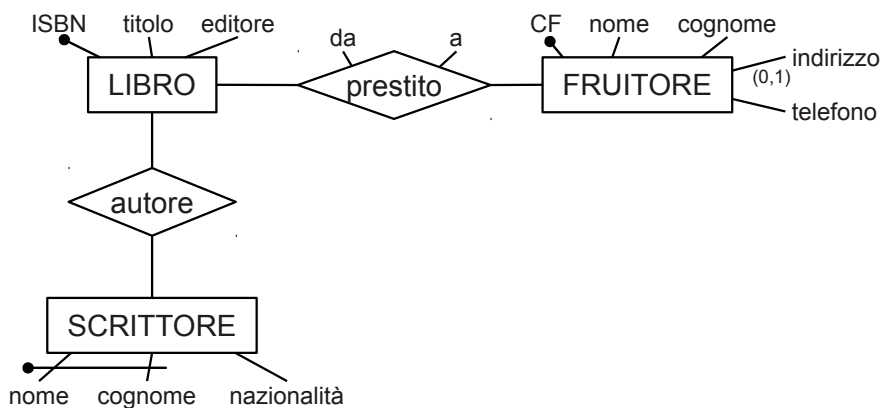
Parte prima

Nota: perché la seconda parte della prova venga corretta è necessario rispondere correttamente ad almeno 3 domande su 5, in caso contrario la prova verrà considerata non superata.

- Elencare i tre vincoli intrarelazionali che possono essere definiti su un attributo in fase di creazione di una tabella.
- In al più tre righe dire quale è la principale differenza tra sistema informativo e sistema informatico.
- Elencare i tre tipi di relazione che possono essere identificati nel modello ER.
- In al più tre righe dire cosa significa che il modello relazionale è basato su valori.
- Elencare i tre possibili metodi per eliminare una generalizzazione da uno schema ER in modo da poterlo tradurre in uno schema relazionale.

Parte seconda

- 1) Considerare il seguente schema ER.



- Aggiungere le cardinalità delle relazioni. [1 punto]
- Cosa significa (0,1) vicino all'attributo indirizzo dell'entità FRUITORE? [1 punto]
- Un fruitore può prendere in prestito (anche contemporaneamente) più libri? Spiegare brevemente (max 3 righe) perché. [1 punto]
- Un fruitore può prendere in prestito lo stesso libro più volte (ovviamente in tempi diversi)? Spiegare brevemente (max 3 righe) perché. [2 punti]
- È possibile avere contemporaneamente nell'entità SCRITTORE le seguenti due istanze?
 - (Carlo, Verdi, Italia)
 - (Carlos, Verdi, Argentina)Spiegare brevemente (max 3 righe) perché. [1 punto]
- Tradurre lo schema ER in uno schema relazionale, indicando in quest'ultimo le chiavi, i vincoli di non nullità e i vincoli di integrità referenziale. [4 punti]

2) Considerare il seguente schema relazionale che rappresenta l'assegnazione del personale dell'Università ad un particolare ufficio/stanza.

- STANZA (numero, piano, edificio, tipo, posti_disponibili*)
- POSTAZIONE (numeroStanza, pianoStanza, edificioStanza, persona)
con vincoli di integrità referenziale tra gli attributi numeroStanza, pianoStanza, edificioStanza e la relazione STANZA e tra l'attributo persona e la relazione PERSONA
- PERSONA (CF, nome, cognome)

e la seguente istanziazione.

STANZA				
numero	piano	edificio	tipo	posti_disponibili
64	1	Ca Vignal 2	ufficio	13
83	1	Ca Vignal 2	ufficio	NULL
1	0	Piramide	laboratorio	120

PERSONA		
CF	nome	cognome
PG	Gabriele	Pozzani
PC	Carla	Parenti
VM	Massimo	Verdi
FF	Franco	Furlan

POSTAZIONE			
numeroStanza	pianoStanza	edificioStanza	persona
64	1	Ca Vignal 2	PC
64	1	Ca Vignal 2	VM
64	1	Ca Vignal 2	FF
64	1	Ca Vignal 2	PG

- Cosa significa l'asterisco accanto all'attributo posti_disponibili? [1 punto]
- Dare i comandi SQL per creare le tre tabelle sopra (incluse eventuali chiavi, vincoli di non nullità e vincoli di integrità referenziale). [2 punti]
- Dare i comandi per inserire la prima riga di ciascuna delle tre tabelle. [2 punti]
- Cosa restituisce la seguente interrogazione? (scrivere la tabella risultante) [2 punti]

```
SELECT numero, piano, edificio, count(persona) as persone
FROM stanza LEFT JOIN postazione ON (numero=numeroStanza AND
piano=pianoStanza AND edificio=edificioStanza)
WHERE tipo = 'ufficio'
GROUP BY numero, piano, edificio
```
- È possibile inserire in POSTAZIONE le seguenti tuple? Per ognuna spiegare brevemente (max 3 righe) perché. [3 punti]
 (64, 1, 'Ca Vignal 2', NULL)
 (64, 1, 'Ca Vignal 2', 'PG')
 (64, 1, 'Ca Vignal 2', 'ZN')

3) Considerare la base di dati definita tramite i seguenti comandi SQL.

```
CREATE TABLE stanza (  
    numero varchar(16) PRIMARY KEY,  
    num_posti integer  
);
```

```
CREATE TABLE orario (  
    da time,  
    a time,  
    PRIMARY KEY (da,a)  
);
```

```
CREATE TABLE speaker (  
    passaporto varchar(16) PRIMARY KEY,  
    nome varchar(32) NOT NULL,  
    cognome varchar(32) NOT NULL,  
    nazione varchar(32)  
);
```

```
CREATE TABLE presentazione (  
    titolo varchar(128) PRIMARY KEY,  
    stanza varchar(16) REFERENCES stanza(numero) ON DELETE NO ACTION ON UPDATE CASCADE,  
    inizio time,  
    fine time,  
    speaker varchar(32) REFERENCES speaker(passaporto) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (inizio, fine) REFERENCES orario(da,a),  
    UNIQUE (inizio, fine, speaker)  
);
```

La base di dati rappresenta le presentazioni durante una conferenza (che si svolge in una sola giornata) con la loro allocazione nel tempo e nelle stanze, il loro titolo e gli speaker che le terranno.

Formulare le interrogazioni SQL che permettano di determinare:

- nome e cognome degli speaker provenienti dalla Cina; [1 punto]
- titolo, stanza e orario delle presentazioni tenute da speaker provenienti dal Brasile; [1 punto]
- il numero di presentazioni che si tengono nella stanza numero 140; [2 punti]
- il titolo delle presentazioni di cui non si conosce ancora la stanza in cui si svolgono; [2 punti]
- il numero di speaker di cui non si conosce la nazione di provenienza; [2 punti]
- il titolo di tutte le presentazioni e, se si conosce, lo speaker (tutti i dati su di esso) che le tiene. [2 punti]

SOLUZIONI

3)

- a)

```
SELECT nome, cognome
FROM speaker
WHERE nazione = 'Cina';
```
- b)

```
SELECT titolo, stanza, inizio, fine
FROM presentazione AS p JOIN speaker AS s ON p.speaker=s.passaporto
WHERE s.nazione = 'Brasile';
```
- c)

```
SELECT count(*)
FROM presentazione
WHERE stanza = '140';
```
- d)

```
SELECT titolo
FROM presentazione
WHERE stanza IS NULL;
```
- e)

```
SELECT count(*)
FROM speaker
WHERE nazione IS NULL;
```
- f)

```
SELECT titolo, s.*
FROM presentazione AS p LEFT JOIN speaker AS s ON p.speaker=s.passaporto
```