

matricola	cognome	nome	firma
-----------	---------	------	-------

A.1 + A.2 + A3	B.1	B.2	B.3	Totale

Istruzioni

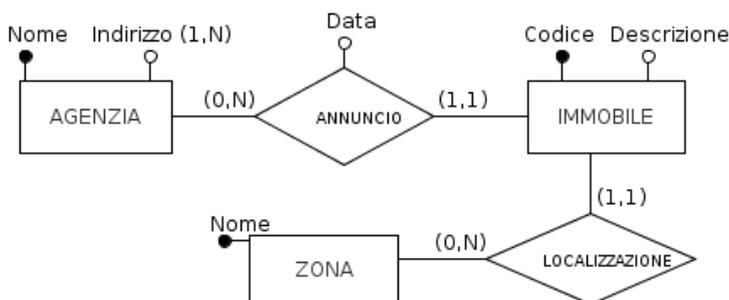
- È vietato portare all'esame libri, eserciziari, appunti e dispense. Chiunque venga trovato in possesso di documentazione relativa al corso – anche se non attinente alle domande proposte – vedrà annullata la propria prova.
- Scrivere solo sui fogli distribuiti, cancellando le parti di brutta con un tratto di penna. Non separare questi fogli.
- Tempo a disposizione: 1 ora e 45 minuti.

A. Parte prima

- A.1. Quali sono le due possibili classificazioni delle generalizzazioni nel modello ER? Descriverle brevemente.
- A.2. Utilizzando il modello Entità-Relazione (ER) fare un esempio di relazione ricorsiva.
- A.3. In al più 10 righe descrivere a cosa servono e quali sono le azioni compensative (o politiche di reazione) alla violazione di un vincolo d'integrità referenziale.

B. Parte seconda

B.1. Considerare il seguente schema ER.



- B.1.a. Spiegare cosa significa (1,N) vicino all'attributo indirizzo.
- B.1.b. Descrivere la relazione che intercorre tra AGENZIA e IMMOBILE, discutendo anche le cardinalità.
- B.1.c. Descrivere l'attributo data presente nel diagramma.
- B.1.d. Se si volesse rappresentare il fatto che un immobile può avere più annunci diversi come si potrebbe modificare l'ER?
- B.1.e. Tradurre lo schema ER in uno schema relazionale (senza tenere conto della modifica del punto precedente), indicando in quest'ultimo eventuali chiavi, vincoli di non nullità e vincoli di integrità referenziale.

B.2. Considerare il seguente schema relazionale che rappresenta una parte di un sistema per la gestione di interventi tecnici:

```

TECNICO(cognome, nome, data_nascita*)
TIPO_INTERVENTO(codice, descrizione)
CLIENTE(cf, nome, indirizzo*)
INTERVENTO(data, cliente, tecnico_cognome, tecnico_nome, tipo_intervento, costo*)
    
```

Con vincoli di integrità referenziale tra cliente in INTERVENTO e cf in CLIENTE e tra tecnico_cognome e tecnico_nome in INTERVENTO e cognome e nome in TECNICO, infine tra tipo_intervento in INTERVENTO e codice in TIPO_INTERVENTO.

Inoltre, è presente un vincolo di tupla che stabilisce che un socio non possa prestare un servizio a se stesso:

CLIENTE		
cf	nome	indirizzo
LGIRSS44S21F861X	Luigi Rossi	Via Verdi 12
MGRNRE71S41L949R	Margherita Neri	NULL
MROBNC69B26A876X	Mario Bianchi	Via Garibaldi 2/A

TIPO_INTERVENTO	
codice	descrizione
1	Riparazione rullo
2	Cambio toner
3	Fornitura carta A4

TECNICO		
cognome	nome	data_nascita
Conti	Luigi	NULL
Conti	Mario	1957-05-09
Super	Mario	1952-11-25
Tube	Franco	NULL

INTERVENTO					
data	cliente	tecnico_cognome	tecnico_nome	tipo_intervento	costo
2013-12-12	LGIRSS44S21F861X	Super	Mario	1	80.90
2013-12-12	LGIRSS44S21F861X	Super	Mario	2	10.20
2013-12-15	MROBNC69B26A876X	Super	Mario	1	10.20
2013-12-16	LGIRSS44S21F861X	Conti	Mario	1	NULL

B.2.a. Scrivere i comandi SQL per creare le tre tabelle sopra riportate (incluse eventuali chiavi, vincoli di non nullità, vincoli di unicità e vincoli di integrità referenziale). Tenere conto che deve essere evitato di cancellare un cliente, un tipo di intervento oppure un tecnico coinvolti in qualche intervento. Le modifiche di chiavi primarie, invece, si “propagano” nel database alle chiavi secondarie.

B.2.b. Cosa restituisce la seguente interrogazione? (scrivere la tabella risultante)

```
SELECT cognome FROM tecnico WHERE cognome LIKE 'C%' ORDER BY cognome;
```

B.2.c. Cosa restituisce la seguente interrogazione? (scrivere la tabella risultante)

```
SELECT cognome, SUM(costo)
FROM tecnico INNER JOIN intervento ON cognome=tecnico_cognome AND nome=tecnico_nome
GROUP BY cognome;
```

B.2.d. Scrivere i comandi SQL per cancellare in maniera corretta e ordinata dal DB il tecnico “Conti Mario” e i suoi interventi.

B.2.e. Dire se i seguenti comandi sono corretti e in tal caso scrivere quanti record avranno dopo la loro esecuzione le tabelle INTERVENTO e CLIENTE sull'istanza sopra riportata nelle figure, oppure se non lo è spiegare il motivo (max 3 righe).

```
DELETE FROM tipo_intervento WHERE codice > 1;
UPDATE tipo_intervento SET codice = 3 WHERE nome = 'Cambio toner';
```

B.3. Si consideri il seguente schema SQL, che rappresenta un semplice sistema di gestione di opere teatrali.

```
CREATE TABLE opera (
    titolo VARCHAR(100) PRIMARY KEY,
    autore VARCHAR(40),
    descrizione VARCHAR(250)
);

CREATE TABLE ruolo (
    nome VARCHAR(40) PRIMARY KEY,
    descrizione VARCHAR(250) NOT NULL,
    opera VARCHAR(100),
    FOREIGN KEY(opera) REFERENCES opera(titolo) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE artista (
    nome VARCHAR(30),
    cognome VARCHAR(30),
    email VARCHAR(30) NOT NULL,
    cellulare VARCHAR(20),
    PRIMARY KEY(nome, cognome)
);

CREATE TABLE assegnazione_ruolo (
    artista_nome VARCHAR(30),
    artista_cognome VARCHAR(30),
    ruolo VARCHAR(40),
    anno INT,
    PRIMARY KEY(artista_nome, artista_cognome, ruolo, anno),
    FOREIGN KEY(artista_nome, artista_cognome) REFERENCES artista(nome, cognome) ON
        DELETE NO ACTION ON UPDATE CASCADE,
    FOREIGN KEY(ruolo) REFERENCES ruolo(nome) ON DELETE CASCADE ON UPDATE CASCADE
);
```

scrivere i comandi SQL che permettono di:

B.3.a. Ottenere la lista ordinata degli artisti, di cui sia noto il numero di cellulare, in ordine decrescente per cognome e nome.

B.3.b. Ottenere la lista dei ruoli, con rispettiva descrizione, e dei rispettivi artisti, con cognome e nome, dell'opera “Romeo e Giulietta”.

B.3.c. Ottenere gli artisti anche con la loro mail, riportata come "e-mail", e il numero dei ruoli assegnati (riportato come "numero_ruoli") a cui sono stati assegnati almeno 2 ruoli, ordinati per cognome e nome.

B.3.d. Aggiornare nell'intero database il titolo dell'opera “Giulietta e Romeo” in “Romeo e Giulietta”.

B.3.e. Rimuovere gli artisti ai quali non è stato assegnato alcun ruolo.