

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

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

**Istruzioni**

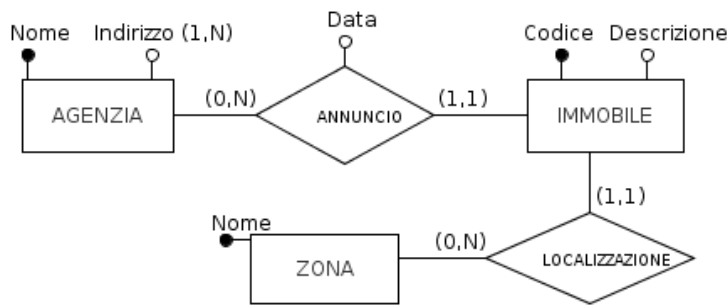
- È vietato portare all'esame libri, eserciziari, appunti e dispense. Chiunque venga trovato in possesso di documentazione relativa al corso – anche se non attinente alle domande proposte – vedrà annullata la propria prova.
- Scrivere solo sui fogli distribuiti, cancellando le parti di brutta con un 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		
<u>cf</u>	<u>nome</u>	<u>indirizzo</u>
LGIRSS44S21F861X	Luigi Rossi	Via Verdi 12
MRGNRE71S41L949R	Margherita Neri	NULL
MROBNC69B26A876X	Mario Bianchi	Via Garibaldi 2/A

TIPO_INTERVENTO	
<u>codice</u>	<u>descrizione</u>
1	Riparazione rullo
2	Cambio toner
3	Fornitura carta A4

TECNICO		
<u>cognome</u>	<u>nome</u>	<u>data_nascita</u>
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.

# SOLUZIONI

## A. Parte prima

- A.1. Totale-Parziale e Esclusiva-Sovrapposta. La prima indica il fatto che l'unione delle entità figlie copre, cioè costituisce e rappresenta, interamente oppure in parte l'insieme dell'entità genitore. Una generalizzazione è esclusiva quando l'intersezione dei sottoinsiemi delle sue entità figlie è vuota, sovrapposta in caso contrario.
- A.2. Una relazione o associazione ricorsiva mette in relazione un'entità con se stessa. Come si può vedere dal libro di testo (Atzeni e altri) nella fig. 7.8, in alcuni casi è necessario stabilire anche dei ruoli che l'entità coinvolta svolge nella relazione.
- A.3. Servono a mantenere una situazione di integrità dei dati, indicando le reazioni che devono essere messe in campo a seguito di un'azione di cancellazione o modifica della base di dati, che vada a violare vincoli di integrità referenziale. In particolare, si può impedire l'azione in oggetto (NO ACTION) oppure propagare ai dati correlati la cancellazione o la modifica (CASCADE). Inoltre, è altresì possibile specificare per i campi delle foreign key coinvolte i valori null o di default (SET NULL e SET DEFAULT).

## B. Parte seconda

### B.1.

B.1.a. L'attributo indirizzo è obbligatorio (1) e un'agenzia può avere più indirizzi.

B.1.b. Tra AGENZIA e IMMOBILE esiste una relazione binaria uno a molti, nella quale un'agenzia può non aver ancora pubblicato alcun annuncio oppure averne pubblicati più d'uno. Al contrario, ogni immobile presente nel database ha associato sempre uno ed un solo annuncio.

B.1.c. Si tratta di un attributo proprio della associazione ANNUNCIO ed indica la data di pubblicazione dell'annuncio stesso da parte dell'agenzia.

B.1.d. Si potrebbe modificare la cardinalità, con cui un immobile partecipa alla associazione ANNUNCIO in (1, N).

B.1.e. AGENZIA(Nome, Indirizzo)

IMMOBILE(Codice, Descrizione\*, Zona, DataAnnuncio, Agenzia)

ZONA(Nome)

INDIRIZZI\_SECONDARI(Indirizzo, Agenzia)

con vincolo di integrità referenziale tra l'attributo Agenzia dell'entità INDIRIZZO e l'entità AGENZIA, tra l'attributo Agenzia di IMMOBILE e l'entità AGENZIA e tra l'attributo Zona di IMMOBILE e l'entità ZONA.

Infine, per gli indirizzi oltre a quello obbligatorio dell'agenzia, ad es. per le filiali, ho un vincolo di integrità referenziale tra l'attributo Agenzia dell'entità INDIRIZZI\_SECONDARI e l'entità AGENZIA.

In alternativa all'includere la data e l'agenzia dell'annuncio in IMMOBILE potrei creare una tabella a parte (coi relativi vincoli di integrità referenziale:

ANNUNCIO(Immobile, Agenzia, Data)

### B.2.

#### B.2.a.

```
CREATE TABLE tecnico (
    cognome VARCHAR(32),
    nome VARCHAR(32),
    data_nascita DATE,
    PRIMARY KEY(cognome, nome)
);

CREATE TABLE tipo_intervento (
    codice INT PRIMARY KEY,
    descrizione VARCHAR(32) NOT NULL
);

CREATE TABLE cliente (
    cf CHAR(16) PRIMARY KEY,
    nome VARCHAR(32) NOT NULL,
    indirizzo VARCHAR(32)
);

CREATE TABLE intervento (
    data DATE,
    cliente CHAR(16),
    tecnico_cognome VARCHAR(32),
    tecnico_nome VARCHAR(32),
    tipo_intervento INT,
    costo DECIMAL(19,2),
    PRIMARY KEY(data, cliente, tecnico_cognome, tecnico_nome, tipo_intervento),
    FOREIGN KEY(cliente) REFERENCES cliente(cf) ON UPDATE CASCADE ON DELETE NO ACTION,
```

```

FOREIGN KEY(tecnico_cognome, tecnico_nome) REFERENCES tecnico(cognome, nome) ON
UPDATE CASCADE ON DELETE NO ACTION,
FOREIGN KEY(tipo_intervento) REFERENCES tipo_intervento(codice) ON UPDATE CASCADE ON
DELETE NO ACTION
);

```

B.2.b.

cognome
Conti
Conti

B.2.c.

cognome	sum
Conti	NULL
Super	101.3000

B.2.d. In base ai vincoli precedenti, per eliminare una squadra devo prima eliminare gli eventuali interventi del tecnico indicato e poi posso procedere alla sua eliminazione.

```

DELETE FROM intervento WHERE tecnico_cognome = 'Conti' AND tecnico_nome = 'Mario';
DELETE FROM tecnico WHERE cognome = 'Conti' AND nome = 'Mario';

```

B.2.e. I comandi violano entrambi dei vincoli di integrità referenziale, determinando i seguenti errori in PostgreSQL:

SQL error:

```

ERROR: update or delete on table "tipo_intervento" violates foreign key constraint
"intervento_tipo_intervento_fkey" on table "intervento"
DETAIL: Key (codice)=(2) is still referenced from table "intervento".

```

```

ERROR: duplicate key value violates unique constraint "tipo_intervento_pkey"
DETAIL: Key (codice)=(3) already exists.

```

Tra l'altro, la seconda istruzione presenta anche l'errore che il campo nome non è presente in tipo\_intervento

B.3.

B.3.a. `SELECT * FROM artista WHERE NOT cellulare IS NULL ORDER BY cognome, nome DESC;`

B.3.b. `SELECT artista_cognome, artista_nome, ruolo, descrizione FROM assegnazione_ruolo INNER JOIN ruolo AS R ON R.nome = ruolo WHERE opera = 'Romeo e Giulietta' ORDER BY artista_cognome, artista_nome;`

-- Oppure

```

SELECT A.cognome, A.nome, ruolo, descrizione FROM assegnazione_ruolo INNER JOIN artista AS
A ON artista_cognome = A.cognome AND artista_nome = A.nome INNER JOIN ruolo AS R ON R.nome
= ruolo WHERE opera = 'Romeo e Giulietta' ORDER BY A.cognome, A.nome;

```

B.3.c. `SELECT cognome, nome, email AS "e-mail", COUNT(*) AS numero_ruoli FROM assegnazione_ruolo INNER JOIN artista ON artista_cognome = cognome AND artista_nome = nome GROUP BY cognome, nome, email HAVING COUNT(*) >= 2 ORDER BY cognome, nome;`

B.3.d. `UPDATE opera SET titolo = 'Giulietta e Romeo' WHERE titolo = 'Romeo e Giulietta';`

B.3.e. `DELETE FROM artista WHERE (nome, cognome) NOT IN (SELECT artista_nome, artista_cognome FROM assegnazione_ruolo);`