

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

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

Istruzioni

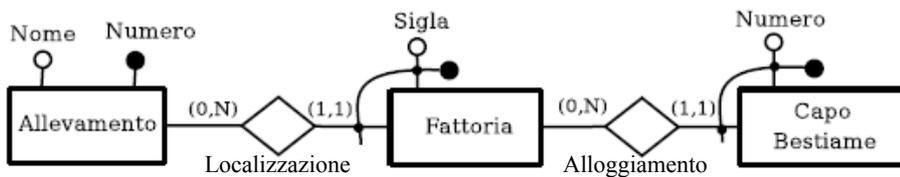
- È vietato portare all'esame libri, eserciziari, appunti e dispense. Chiunque venga trovato in possesso di documentazione relativa al corso – anche se non attinente alle domande proposte – vedrà annullata la propria prova.
- Scrivere solo sui fogli distribuiti, cancellando le parti di brutta con un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

A. Parte prima

- A.1. In al più 5 righe descrivere quali situazioni può rappresentare in una base di dati il valore NULL.
 A.2. In al più 5 righe descrivere cosa si intende per componente intensionale e componente estensionale di una base di dati.
 A.3. In al più 5 righe spiegare il concetto di atomicità di una transazione e fornirne un semplice esempio.

B. Parte seconda

B.1. Considerare il seguente schema ER.



- B.1.a. In al più 3 righe, spiegare che tipo di entità è FATTORIA e quale è la sua chiave.
 B.1.b. In al più 3 righe, spiegare cosa significa (1,1) vicino all'entità Fattoria.
 B.1.c. Solo sulla base dell'ER (cioè senza ulteriori vincoli e/o informazioni) è possibile per una stessa fattoria mettere più capi di bestiame? Spiegare brevemente (max 3 righe) perché.
 B.1.d. All'entità Allevamento aggiungere, sullo schema sopra disegnato, un attributo facoltativo anno_creazione, che vuole rappresentare l'anno in cui l'allevamento è stato creato.
 B.1.e. Tradurre lo schema ER in uno schema relazionale, indicando in quest'ultimo eventuali chiavi, vincoli di non nullità e vincoli di integrità referenziale.

B.2. Considerare il seguente schema relazionale che rappresenta un sistema di prestiti bibliotecari:

LIBRO(ISBN, titolo, nome_autore, cognome_autore, pagine*)

UTENTE(CF, nome, cognome, eta*)

PRESTITO(libro, utente, data_prestito, data_restituzione*)

con vincoli di integrità referenziale tra l'attributo libro di PRESTITO con la relazione LIBRO e tra l'attributo utente di PRESTITO con la relazione UTENTE, e la seguente istanza.

LIBRO				
isbn	titolo	nome_autore	cognome_autore	pagine
X16	La fattoria degli animali	George	Orwell	125
A51	1984	George	Orwell	175
L86	Non avere paura dei libri	Christian	Mascheroni	NULL
L31	Lucky Starr e i pirati degli asteroidi	Isaac	Asimov	154

UTENTE			
cf	nome	cognome	eta
CRLRSS	Carlo	Rossi	44
CHRBNC	Chiara	Bianchi	NULL

PRESTITO			
libro	utente	data_prestito	data_restituzione
X16	CRLRSS	17-02-2013	28-02-2013
A51	CHRBNC	18-02-2013	NULL
X16	CRLRSS	19-02-2013	NULL
L31	CRLRSS	19-02-2013	13-04-2013

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 una utente o un libro, a cui fa riferimento almeno una tupla di PRESTITO, e che in caso di modifica del codice ISBN di un libro, tale modifica si deve riflettere in automatico anche in PRESTITO.

B.2.b. Scrivere il comando SQL per inserire in LIBRO il libro “I pirati della Malesia” di Emilio Salgari, il cui numero di pagine è sconosciuto.

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

```
SELECT cognome_autore, nome_autore, avg(pagine) AS media_pagine
FROM LIBRO
WHERE cognome_autore = 'Orwell'
GROUP BY cognome_autore, nome_autore
```

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

```
SELECT *
FROM UTENTE
WHERE eta IS NOT NULL
```

B.2.e. Dire se il seguente comando è corretto e in tal caso scrivere la tabella risultante dalla sua esecuzione sull'istanza sopra riportata, oppure se non lo è spiegare il motivo (max 3 righe).

```
INSERT INTO UTENTE VALUES ('Carlo', 'Rossi')
```

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

```
CREATE TABLE aeroporto(
    citta varchar(50) PRIMARY KEY,
    nazione varchar(50) NOT NULL,
    num_piste integer
);
```

```
CREATE TABLE aereo(
    tipo_aereo varchar(30) PRIMARY KEY,
    num_passeggeri char(16),
    qta_merci integer
);
```

```
CREATE TABLE volo(
    idvolo integer PRIMARY KEY,
    giorno_settimana varchar(12) NOT NULL,
    citta_partenza varchar(50),
    ora_partenza time,
    citta_arrivo varchar(50),
    ora_arrivo time,
    tipo_aereo varchar(30),
    foreign key(citta_partenza) REFERENCES aeroporto(citta) ON UPDATE CASCADE,
    foreign key(citta_arrivo) REFERENCES aeroporto(citta) ON UPDATE CASCADE,
    foreign key(tipo_aereo) REFERENCES aereo(tipo_aereo) ON UPDATE CASCADE
);
```

scrivere le interrogazioni SQL che permettono di determinare:

B.3.a. Le città con un aeroporto di cui non è noto il numero di piste.

B.3.b. Le nazioni da cui parte e arriva il volo con codice AZ274.

B.3.c. I tipi di aereo e il corrispondente numero di passeggeri per i tipi di aereo usati nei voli che partono da Torino. Se il numero dei passeggeri dell'aereo non è disponibile, visualizzare solamente il tipo.

B.3.d. Il numero di voli internazionali che partono il giovedì da Napoli.

B.3.e. Le città francesi da cui partono più di venti voli alla settimana diretti in Italia.