

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

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

**Istruzioni**

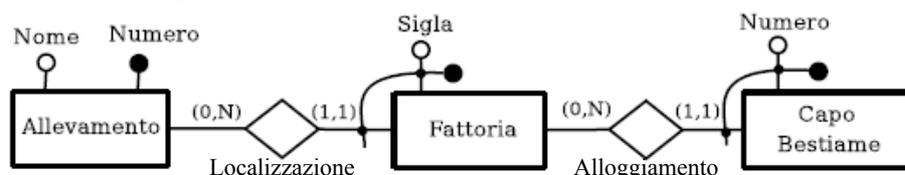
- È vietato portare all'esame libri, eserciziari, appunti e dispense. Chiunque venga trovato in possesso di documentazione relativa al corso – anche se non attinente alle domande proposte – vedrà annullata la propria prova.
- Scrivere solo sui fogli distribuiti, cancellando le parti di brutta con un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

**A. Parte prima**

- A.1. 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.

## SOLUZIONI

### A. Parte prima

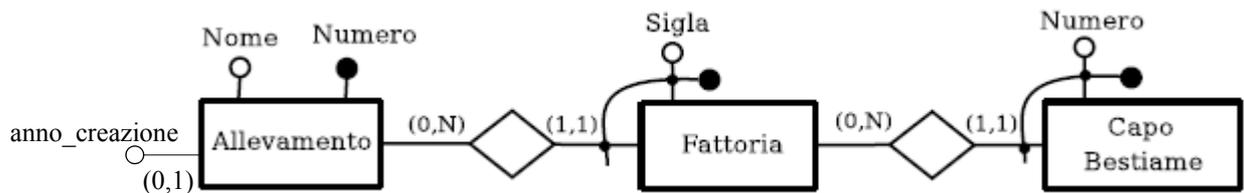
- A.1. Il valore NULL può rappresentare: un valore sconosciuto, un valore inesistente oppure un valore di cui non si ha alcuna informazione, il valore potrebbe esistere o non esistere e se esiste non sappiamo quale sia.
- A.2. Per componente intensionale di una base di dati si intende la sua struttura, il suo schema di descrizione dei dati e delle loro relazioni, mentre per componente estensionale o istanza ci riferiamo ai dati veri e propri, cioè ai valori presenti nel database in un dato istante.
- A.3. Con atomicità di una transazione indichiamo la caratteristica per cui una sequenza di comandi con “esecuzione indivisibile” su una base di dati: o viene eseguita del tutto o per nulla (non è ammessa l'esecuzione parziale).  
Possiamo pensare ad una transazione bancaria con il passaggio di denaro da un conto corrente ad un altro. Perché la transazione sia consistente prelievo e versamento devono avvenire entrambi oppure nessuno dei due.

### B. Parte seconda

B.1.

- B.1.a. L'entità FATTORIA è un'entità debole e il suo identificatore esterno è dato dal proprio attributo “sigla” e dall'attributo “numero” dell'entità ALLEVAMENTO.
- B.1.b. Che ogni occorrenza di Fattoria partecipa alla associazione con allevamento con ciascuna occorrenza e una e una sola volta, cioè una fattoria è localizzata in uno e un solo allevamento.
- B.1.c. Poiché la cardinalità della partecipazione di Fattoria all'associazione Alloggiamento con “Capo Bestiame” è N, una fattoria può alloggiare più capi di bestiame.

B.1.d.



B.1.e.

```
ALLEVAMENTO(Numero, Nome*, anno_creazione*)
FATTORIA(Sigla, Numero Allevamento)
CAPO BESTIAME(Sigla Fattoria, Numero Allevamento, Numero CapoBestiame)
```

con vincoli di integrità referenziale tra l'attributo Numero\_Allevamento di Fattoria e Numero di Allevamento, Numero\_Allevamento di Capo bestiame e sempre Numero di Allevamento e tra Sigla\_Fattoria di Capo Bestiame e Sigla di Fattoria.

B.2.

B.2.a.

```
CREATE TABLE libro (
    isbn varchar(16) PRIMARY KEY,
    titolo varchar(64) NOT NULL,
    nome_autore varchar(32) NOT NULL,
    cognome_autore varchar(32) NOT NULL,
    pagine integer
);

CREATE TABLE utente (
    cf char(16) PRIMARY KEY,
    nome varchar(32) NOT NULL,
    cognome varchar(32) NOT NULL,
    eta integer
);

CREATE TABLE prestito (
    libro varchar(16) REFERENCES libro(isbn)
    ON DELETE NO ACTION ON UPDATE CASCADE,
```

```

utente char(16) REFERENCES utente(cf)
      ON DELETE NO ACTION ON UPDATE CASCADE, -- ON UPDATE facoltativo
data_prestito date,
data_restituzione date,
PRIMARY KEY (libro, utente, data_prestito)
);

```

B.2.b. Non posso inserire in LIBRO il libro “I pirati della Malesia” di Emilio Salgari senza avere a disposizione il suo codice ISBN, per cui o concludo dicendo che la cosa non è possibile oppure ne invento uno all'uopo, ad es.

```

INSERT INTO libro(isbn, titolo, nome_autore, cognome_autore) VALUES('S01', 'I
pirati della Malesia', 'Emilio', 'Salgari')

```

B.2.c.

<b>cognome_autore</b>	<b>nome_autore</b>	<b>media_pagine</b>
Orwell	George	150

B.2.d.

<b>cf</b>	<b>nome</b>	<b>cognome</b>	<b>eta</b>
CRLRS	Carlo	Rossi	44

B.2.e. Il comando non è corretto poiché non si può inserire un utente senza codice fiscale, che rappresenta la chiave primaria della tabella UTENTE ed è quindi è un attributo obbligatorio. Inoltre, non essendo indicati i nomi degli attributi dovrei avere un valore per ciascuno degli attributi della tabella.

B.3.

B.3.a. `SELECT citta FROM aeroporto WHERE num_piste IS NULL`

B.3.b. C'è un'incongruenza nel testo perchè il codice a cui si riferisce la domanda è di tipo VARCHAR mentre idvolo è INTEGER, supponendo quindi l'attributo di tipo VARCHAR abbiamo la seguente interrogazione (chiaramente anche soluzioni analoghe visto l'errore sono ritenute accettabili):

```

SELECT AP.nazione AS nazione_partenza, AA.nazione AS nazione_arrivo
FROM aeroporto AS AP JOIN volo ON citta_partenza = AP.citta
JOIN aeroporto AS AA ON citta_arrivo = AA.citta
WHERE idvolo = 'AZ274'

```

B.3.c. `SELECT aereo.tipo_aereo, aereo.num_passeggeri
FROM aereo JOIN volo ON aereo.tipo_aereo = volo.tipo_aereo
WHERE citta_partenza = 'Torino'`

B.3.d. `SELECT COUNT(*)
FROM volo JOIN aeroporto ON citta_arrivo = citta
WHERE nazione <> 'Italia'
AND giorno_settimana = 'giovedì'
AND citta_partenza = 'Napoli';`

oppure:

```

SELECT COUNT(idvolo) FROM volo
WHERE giorno_settimana = 'giovedì'
AND citta_partenza = 'Napoli'
AND citta_arrivo IN (
SELECT citta FROM aeroporto WHERE nazione <> 'Italia')

```

B.3.e. `SELECT citta_partenza
FROM volo JOIN aeroporto AS AP ON citta_partenza = AP.citta
JOIN aeroporto AS AA ON citta_arrivo = AA.citta
WHERE AP.nazione = 'Francia'
AND AA.nazione = 'Italia'
GROUP BY citta_partenza
HAVING COUNT(*) > 20;`