

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

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

**Istruzioni**

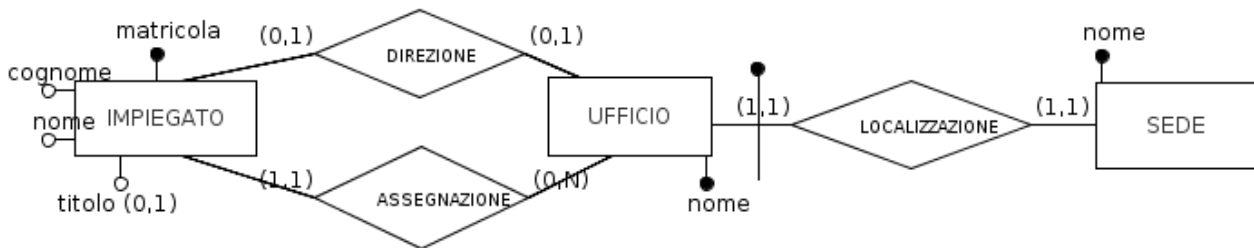
- È vietato portare all'esame libri, eserciziari, appunti e dispense. Chiunque venga trovato in possesso di documentazione relativa al corso – anche se non attinente alle domande proposte – vedrà annullata la propria prova.
- Scrivere solo sui fogli distribuiti, cancellando le parti di brutta con un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

**A. Parte prima**

- A.1. In al più 10 righe descrivere quali vincoli esprimono le cardinalità nel modello E/R.  
 A.2. In al più 5 righe descrivere la differenza nei sistemi informativi tra informazione e dato, Fornire un esempio al riguardo.  
 A.3. In al più 5 righe descrivere che cos'è un DBMS.

**B. Parte seconda**

B.1. Considerare il seguente schema ER.



- B.1.a. Spiegare cosa significa (0,1) vicino all'attributo titolo.  
 B.1.b. Descrivere la relazione che intercorre tra IMPIEGATO e UFFICIO, discutendo anche le cardinalità.  
 B.1.c. Descrivere l'identificatore dell'entità UFFICIO e in particolare dire se la cardinalità verso LOCALIZZAZIONE è necessario che sia (1,1) o potrebbe essere diversa.  
 B.1.d. Se si volesse rappresentare anche la data di assegnazione di un impiegato ad un ufficio come si potrebbe modificare l'ER?  
 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 una parte di un sistema per la gestione di :

ATLETA(nome, cognome, data\_nascita\*, squadra)  
 SQUADRA(nome, anno\_fondazione\*, citta)  
 PARTITA(squadra1, squadra2, data)  
 CITTA(nome)

con vincoli di integrità referenziale tra l'attributo squadra di ATLETA e l'attributo nome di SQUADRA, tra l'attributo citta di SQUADRA e l'attributo nome di CITTA e infine tra l'attributo squadra1 di PARTITA e l'attributo nome di SQUADRA e tra l'attributo squadra2 di PARTITA e l'attributo nome di SQUADRA, e la seguente istanza:

ATLETA			
nome	cognome	data_nascita	squadra
Bruno	Conti	NULL	Roma
Fulvio	Collovati	1957-05-09	Inter
Gabriele	Orioli	1952-11-25	Inter
Gaetano	Scirea	NULL	Juventus
Dino	Zoff	1942-02-28	Juventus
Antonio	Cabrini	1957-10-08	Juventus

SQUADRA		
nome	anno_fondazione	citta
Juventus	1897	Torino
Inter	NULL	Milano
Roma	1927	Roma

CITTA
nome
Milano
Torino
Roma

PARTITA		
squadra1	squadra2	data
Juventus	Inter	2013-03-12
Roma	Juventus	2013-03-19
Roma	Inter	NULL
Inter	Juventus	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 una squadra coinvolta in qualche partita o nella quale milita un atleta. Stesso discorso per quanto riguarda l'eliminazione di una città. 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 atleta WHERE cognome LIKE 'C%' ORDER BY cognome;
```

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

```
SELECT S.nome, S.citta
FROM squadra AS S INNER JOIN partita AS P ON S.nome = P.squadral
WHERE S.nome IN (SELECT squadra FROM atleta GROUP BY squadra HAVING COUNT(*) > 1);
```

B.2.d. Scrivere i comandi SQL per cancellare dal database la squadra Juventus con le relative partite.

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

```
UPDATE squadra SET nome = 'Roma AS' WHERE nome = 'Roma';
DELETE FROM partita WHERE data IS NULL AND squadral <> 'Roma AS';
```

B.3. Si consideri il seguente schema SQL, che rappresenta un semplice sistema di gestione ordini di articoli da parte di clienti.

```
CREATE TABLE cliente (
    nome VARCHAR(30) PRIMARY KEY,
    indirizzo VARCHAR(30) NOT NULL,
    citta VARCHAR(30) NOT NULL
);

CREATE TABLE articolo (
    codice VARCHAR(10) PRIMARY KEY,
    descrizione VARCHAR(30) NOT NULL,
    costo_unitario DECIMAL(10,2) -- Currency in euro with cents
);

CREATE TABLE ordine (
    numero INT PRIMARY KEY,
    cliente VARCHAR(10) NOT NULL,
    data DATE,
    FOREIGN KEY(cliente) REFERENCES cliente(nome) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE dettaglio_ordine (
    numero_ordine INT,
    codice_articolo VARCHAR(10),
    quantita INT NOT NULL DEFAULT 1,
    PRIMARY KEY(numero_ordine, codice_articolo),
    FOREIGN KEY(numero_ordine) REFERENCES ordine(numero) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(codice_articolo) REFERENCES articolo(codice) ON DELETE CASCADE ON UPDATE CASCADE
);
```

scrivere i comandi SQL che permettono di:

B.3.a. Ottenere la lista ordinata dei clienti in ordine crescente per nome.

B.3.b. Ottenere il costo totale per gli ordini numero 1 e 3, distinta ciascun ordine.

B.3.c. Ottenere la lista, con numero e cliente, degli ordini, che siano stati effettuati dopo il 31 dicembre 2012 oppure per i quali non sia specificata la data.

B.3.d. Aggiornare nell'intero database il cliente ('ACME srl', 'Via Garibaldi 22', 'Verona'), che si è trasferito in Via Meucci 12 a Roma e il cui nome è ora 'ACME Spa'.

B.3.e. Rimuovere gli ordini effettuati da clienti che hanno effettuato meno di 2 ordini.