

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

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

Istruzioni

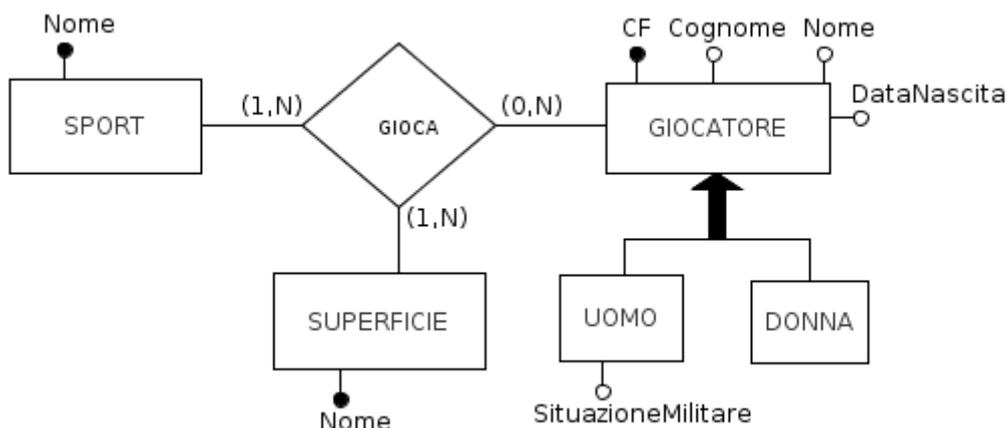
- È vietato portare all'esame libri, eserciziari, appunti e dispense. Chiunque venga trovato in possesso di documentazione relativa al corso – anche se non attinente alle domande proposte – vedrà annullata la propria prova.
- Scrivere solo sui fogli distribuiti, cancellando le parti di brutta con un 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 la differenza nei sistemi informativi tra *informazione* e *dato*.
- A.2. In al più 5 righe in riferimento ai database descrivere la progettazione concettuale e qual è il suo scopo.
- A.3. In al più 5 righe spiegare cos'è la chiave primaria nel modello relazionale. La sua esistenza è sempre garantita? Perché?

B. Parte seconda

B.1. Considerare il seguente schema ER.



- B.1.a. In al più 5 righe, descrivere e classificare la generalizzazione in figura.
- B.1.b. In al più 3 righe, spiegare cosa significa (0,N) vicino all'entità GIOCATORE.
- B.1.c. Riportare il diagramma sul proprio foglio e aggiungere le cardinalità mancanti.
- B.1.d. In al più 5 righe dire se è possibile rappresentare il fatto che un giocatore non possa giocare ad un determinato sport, spiegando brevemente il perché della propria risposta.
- 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 scolastico:

CLASSE(nome_sezione, anno_sezione, piano*)

DOCENTE(cf, nome, cognome, data_nascita*)

INSEGNAMENTO(insegnante, nome_sezione, anno_sezione, materia)

con vincoli di integrità referenziale tra l'attributo insegnante di INSEGNAMENTO con la relazione DOCENTE e tra gli attributi nome_sezione e anno_sezione di INSEGNAMENTO con la relazione CLASSE, e la seguente istanza.

CLASSE		
nome_sezione	anno_sezione	piano
A	1	primo
B	1	terra
A	2	terra
A	3	primo

DOCENTE			
cf	nome	cognome	data_nascita
CRLRSS69B26F861J	Carlo	Rossi	26/02/1969
CHRBNC74S01L949K	Chiara	Bianchi	01/11/1974
LGUNT64b19L949K	Luigi	Inutile	NULL

INSEGNAMENTO			
<u>insegnante</u>	<u>nome_sezione</u>	<u>anno_sezione</u>	<u>materia</u>
CRLRSS69B26F861J	A	1	Informatica
CHRBNC74S01L949K	B	1	Informatica
CHRBNC74S01L949K	A	2	Informatica
CRLRSS69B26F861J	A	3	Sistemi
CHRBNC74S01L949K	A	3	Informatica

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 docente o una classe, a cui fa riferimento almeno una tupla di INSEGNAMENTO, e che in caso di modifica del codice fiscale di un docente, tale modifica si deve riflettere in automatico anche in INSEGNAMENTO.

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

```
SELECT piano, COUNT(*) AS numero_classi
FROM classe
GROUP BY piano
```

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

```
SELECT *
FROM insegnamento
WHERE nome_sezione = 'A' AND insegnante LIKE '%L949K'
```

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

```
DELETE FROM insegnamento WHERE nome_sezione = 'B'
```

B.2.e. Scrivere il comando SQL per aggiungere alla tabella CLASSE la “seconda B”, a cui non è ancora stato assegnata un'aula in un certo piano.

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

```
CREATE TABLE nazione(
    nome VARCHAR(40) PRIMARY KEY
);
```

```
CREATE TABLE autore(
    nome VARCHAR(20),
    cognome VARCHAR(20),
    data_nascita DATE,
    nazione_nascita VARCHAR(40) NOT NULL,
    PRIMARY KEY(nome, cognome),
    FOREIGN KEY(nazione_nascita) REFERENCES nazione(nome)
);
```

```
CREATE TABLE libro(
    titolo VARCHAR(40) PRIMARY KEY,
    nome_autore VARCHAR(20),
    cognome_autore VARCHAR(20),
    lingua VARCHAR(20),
    FOREIGN KEY(nome_autore, cognome_autore) REFERENCES autore(nome, cognome)
    ON DELETE CASCADE
    ON UPDATE SET NULL
);
```

scrivere i comandi SQL che permettono di:

B.3.a. Ottenere il numero totale dei libri in italiano e il numero totale in portoghese.

B.3.b. Ottenere il cognome e il nome, in ordine alfabetico, degli autori nati in Italia, anche se non hanno pubblicato alcun libro.

B.3.c. Ottenere per tutte le nazioni il relativo numero di autori presenti.

B.3.d. Aggiungere il libro in lingua originale dello scrittore portoghese José Saramago 'Memorial do Convento', tenendo conto che non sono ancora presenti nel database né il Portogallo nella tabella nazioni né José Saramago, di cui non si conosce la data di nascita, tra gli autori.

B.3.e. Rimuovere i libri degli scrittori, di cui non si conosce la data di nascita e di cui esiste almeno un libro in italiano.

SOLUZIONI

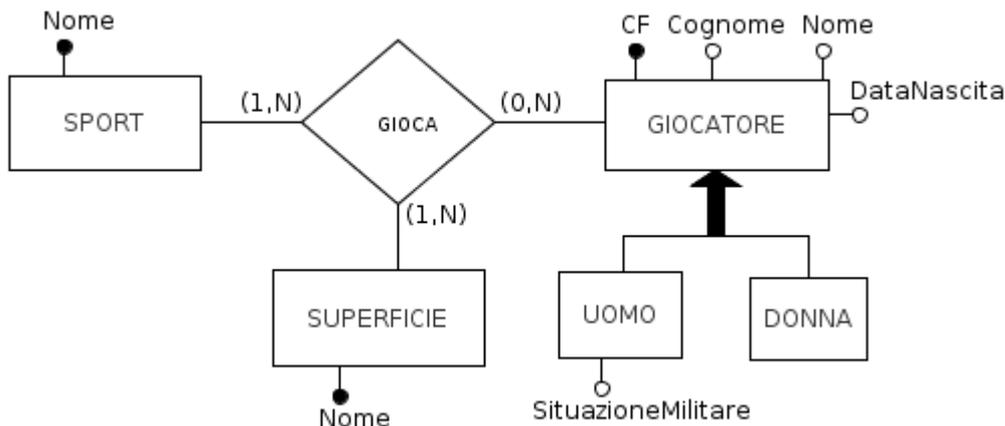
A. Parte prima

- A.1. Possiamo pensare ad un dato come ad un dato “grezzo”, cioè non elaborato, non interpretato, a all'informazione come al dato interpretato (o elaborato), che arricchisce la conoscenza dell'interprete. Ad esempio il dato numerico 37845983 diventa un'informazione nel momento in cui lo interpreto ad es. come numero di telefono di una certa persona oppure diventa un'altra informazione se lo interpreto come il numero di abitanti della Spagna al 31/12/1985..
- A.2. La progettazione concettuale rappresenta la fase iniziale della progettazione di una base di dati (poi abbiamo la progettazione logica e quella fisica) ed evidenzia i concetti essenziali, a livello di dati, che interessano la realtà che dobbiamo modellare. Alla fine dovrà produrre uno schema concettuale formalizzato e facilmente comprensibile. Nel modello E-R, i costrutti principali, le entità e le associazioni, sono appunto la modalità attraverso la quale raggruppare e mettere in relazione i concetti di base della realtà considerata.
- A.3. La chiave primaria è un insieme di attributi che permettono di identificare univocamente le tuple di una relazione o tabella. L'esistenza della chiave primaria, che al più è formata da tutti gli attributi o campi dato della relazione, è garantita dall'unicità delle n-uple (o tuple) individuate da una relazione tra uno o più insiemi.

B. Parte seconda

B.1.

- B.1.a. Tra le entità GIOCATORE e UOMO e DONNA abbiamo una generalizzazione totale ed esclusiva in quanto ogni occorrenza dell'entità genitore è una occorrenza di una e una solo delle entità figlie.
- B.1.b. La cardinalità (0, N) in esame indica che ciascuna occorrenza dell'entità GIOCATORE può partecipare o meno all'associazione GIOCA (partecipazione opzionale) ed può partecipare anche più volte. Quindi un giocatore può non giocare (ancora) oppure giocare a più sport e/o su più superfici di gioco.
- B.1.c. La cardinalità di SPORT è (0, N), ma è accettabile anche (1, N); quella di SUPERFICIE è (1, N), tenendo conto che uno sport, come ad es. il tennis, può essere praticato su diverse superfici (terra, erba,...).



B.1.d. Questo vincolo non è rappresentabile tramite i costrutti del modello E-R, può altresì essere aggiunto come nota documentativa a parte.

B.1.e.

```
SPORT(Nome)
SUPERFICIE(Nome)
GIOCATORE(CF, Cognome, Nome, DataNascita*, Sesso, SituazioneMilitare*)
GIOCA(Sport, Giocatore, Superficie)
```

con vincoli di integrità referenziale tra l'attributo Sport di GIOCA con l'attributo Nome di SPORT, l'attributo Giocatore di GIOCA con l'attributo CF di GIOCATORE e l'attributo Superficie sempre di GIOCA con l'attributo Nome di SUPERFICIE.

La scelta degli attributi opzionali, dal momento che non sono riportati sullo schema E-R, è personale.

B.2.

```
B.2.a. CREATE TABLE classe (
        nome_sezione char(1),
        anno_sezione integer,
        piano varchar(32),
        PRIMARY KEY(nome_sezione, anno_sezione)
);
```

```

CREATE TABLE docente (
    cf char(16) PRIMARY KEY,
    nome varchar(32) NOT NULL,
    cognome varchar(32) NOT NULL,
    data_nascita date
);

CREATE TABLE insegnamento (
    insegnante varchar(16),
    nome_sezione char(1),
    anno_sezione integer,
    materia varchar(40),
    PRIMARY KEY (insegnante, nome_sezione, anno_sezione),
    FOREIGN KEY (insegnante) REFERENCES docente(cf) ON DELETE NO ACTION ON
UPDATE CASCADE,
    FOREIGN KEY (nome_sezione, anno_sezione) REFERENCES
classe(nome_sezione, anno_sezione) ON DELETE NO ACTION ON UPDATE CASCADE
);

```

B.2.b.

piano	numero_classi
terra	2
primo	2

B.2.c.

insegnante	nome_sezione	anno_sezione	materia
CHRBNC74S01L949K	A		2 Informatica
CHRBNC74S01L949K	A		3 Informatica

B.2.d. Il comando è corretto e dopo la sua esecuzione la tabella insegnamento avrà 4 record.

B.2.e. INSERT INTO classe(nome_sezione, anno_sezione, piano) VALUES ('B', 2, NULL);
oppure
INSERT INTO classe VALUES ('B', 2, NULL);

B.3.

B.3.a. SELECT lingua, count(*)
FROM libro
WHERE lingua IN ('italiano', 'portoghese')
GROUP BY lingua;

B.3.b. SELECT cognome, nome
FROM autore
WHERE nazione_nascita = 'Italia'
ORDER BY cognome, nome;

B.3.c. SELECT nazione.nome AS nome_nazione, count(nazione_nascita)
FROM nazione LEFT JOIN autore ON nazione.nome = autore.nazione_nascita
GROUP BY nome_nazione;

B.3.d. INSERT INTO nazione VALUES('Portogallo');
INSERT INTO autore(nome, cognome, data_nascita, nazione_nascita) VALUES('José',
'Saramago', NULL, 'Portogallo');
INSERT INTO libro VALUES('Memorial do Convento', 'José', 'Saramago',
'portoghese');

B.3.e. DELETE FROM libro
WHERE titolo IN (
SELECT titolo
FROM libro INNER JOIN autore ON cognome_autore = cognome AND nome_autore = nome
WHERE lingua = 'italiano' AND data_nascita IS NULL);