

**Basi di dati I — 11 luglio 2014**  
**Tempo a disposizione: un'ora e quarantacinque minuti.**

Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

**Domanda 1** (15%) Considerare il seguente schema di relazione:

R(CodiceCorso, Anno, TitoloCorso, CodiceProf, NomeProf, Dipartimento, Indirizzo)

che si riferisce ad un insieme di corsi, tenuti in più anni presso una sede universitaria; in particolare

- CodiceProf identifica un professore, che ha un nome (l'attributo NomeProf) e afferisce ad un Dipartimento (che a sua volta ha un Indirizzo)
- CodiceCorso identifica un corso (il cui nome è l'attributo TitoloCorso); ogni anno un corso ha un solo professore (che può cambiare da un anno all'altro).

Con riferimento a tale schema:

1. specificare chiavi e dipendenze funzionali definite sulla relazione e indicare quali dipendenze causano violazioni della BCNF

2. mostrare una buona decomposizione che soddisfi la BCNF (indicare gli schemi delle relazioni, con le relative chiavi)

**Domanda 2** (10%) Mostrare uno schema ER che descriva la realtà di interesse illustrata nella domanda precedente.

**Domanda 3** (30%) Mostrare lo schema concettuale per una base di dati per un programma di concerti, secondo le seguenti specifiche.

- Ogni concerto ha un codice, un titolo e una descrizione, ed è costituito da una sequenza di pezzi musicali diversi (interessa l'ordine)
- Ogni pezzo ha un codice, un titolo e un autore (con codice e nome); uno stesso pezzo può essere rappresentato in diversi concerti.
- Ogni concerto è eseguito da un'orchestra; ogni orchestra ha un nome (che la identifica), un anno di fondazione e un insieme di orchestrali.
- Ogni orchestrale ha una matricola (univoca nell'ambito della base di dati), nome e cognome, può partecipare a più orchestre, in ciascuna delle quali suona uno e un solo strumento, ma in orchestre diverse può suonare strumenti diversi.
- Ogni strumento ha un codice e un nome
- Ogni concerto è tenuto più volte, in giorni diversi e in orari che possono variare

**Domanda 4** (25%) Considerare la seguente base di dati relazionale:

- PRODOTTI(Codice, Descrizione, Marca)
- VENDITE(Prodotto, Mese, Anno, Quantità)  
con vincolo di integrità referenziale fra Prodotto e la relazione PRODOTTI

Con riferimento a tale base di dati (supponendo che, per i mesi senza vendite, i valori delle quantità siano pari a zero e quindi tutti i mesi di interesse siano riportati):

1. formulare in algebra relazionale l'interrogazione che trova codice, descrizione e marca per ogni prodotto che abbia almeno una vendita nell'anno 2013.

2. formulare in SQL la stessa interrogazione del punto precedente.

3. formulare in SQL l'interrogazione che trova i prodotti che in un certo mese del 2014 hanno avuto vendite inferiori a quelle dello stesso mese del 2013; mostrare il codice e la descrizione del prodotto e le due quantità.

4. formulare in SQL l'interrogazione che trova le coppie di prodotti di una stessa marca; mostrare i due codici, le due descrizioni e la marca.

**Domanda 5** (20%) Considerare una relazione sullo schema:

- PARTITE(ID, Data, Squadra1, Squadra2, Reti1, Reti2)

e una vista definita su di essa, con la seguente istruzione SQL

```
CREATE VIEW partitebis AS
    SELECT data, squadra1 AS squadra, squadra2 AS avversaria, reti1 AS rfatte, reti2 AS rsubite
    FROM partite
UNION
    SELECT data, squadra2 AS squadra, squadra1 AS avversaria, reti2 AS rfatte, reti1 AS rsubite
    FROM partite
```

La figura seguente mostra parte di un'istanza della relazione PARTITE e della corrispondente vista PARTITEBIS.

PARTITE					
ID	Data	Squadra1	Squadra2	Reti1	Reti2
1	12/06	Brasile	Croazia	3	1
2	13/06	Messico	Camerun	1	0
3	13/06	Spagna	Olanda	1	5
...	...	...	...	...	...

PARTITEBIS				
Data	Squadra	Avversaria	RFatte	RSubite
12/06	Brasile	Croazia	3	1
12/06	Croazia	Brasile	1	3
13/06	Messico	Camerun	1	0
13/06	Camerun	Messico	0	1
13/06	Spagna	Olanda	1	5
13/06	Olanda	Spagna	5	1
...	...	...	...	...

1. Formulare in SQL l'interrogazione che genera, per ogni squadra che ha vinto almeno una partita, il numero di partite vinte (nota: l'indicazione di ignorare le squadre che non hanno vinto partite semplifica l'interrogazione; si consiglia di utilizzare la vista PARTITEBIS, che semplifica la scrittura dell'interrogazione)

2. Si vuole produrre la "classifica" indicando, per ogni squadra il numero di partite vinte, pareggiate e perse (sempre con riferimento alle squadre che hanno almeno una partita vinta, una pareggiata e una persa). Descrivere la struttura dell'interrogazione SQL da utilizzare allo scopo, anche senza scrivere tutto il codice.

Possibili soluzioni

Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

**Domanda 1** (15%) Considerare il seguente schema di relazione:

R(CodiceCorso, Anno, TitoloCorso, CodiceProf, NomeProf, Dipartimento, Indirizzo)

che si riferisce ad un insieme di corsi, tenuti in più anni presso una sede universitaria; in particolare

- CodiceProf identifica un professore, che ha un nome (l'attributo NomeProf) e afferisce ad un Dipartimento (che a sua volta ha un Indirizzo)
- CodiceCorso identifica un corso (il cui nome è l'attributo TitoloCorso); ogni anno un corso ha un solo professore (che può cambiare da un anno all'altro).

Con riferimento a tale schema:

1. specificare chiavi e dipendenze funzionali definite sulla relazione e indicare quali dipendenze causano violazioni della BCNF

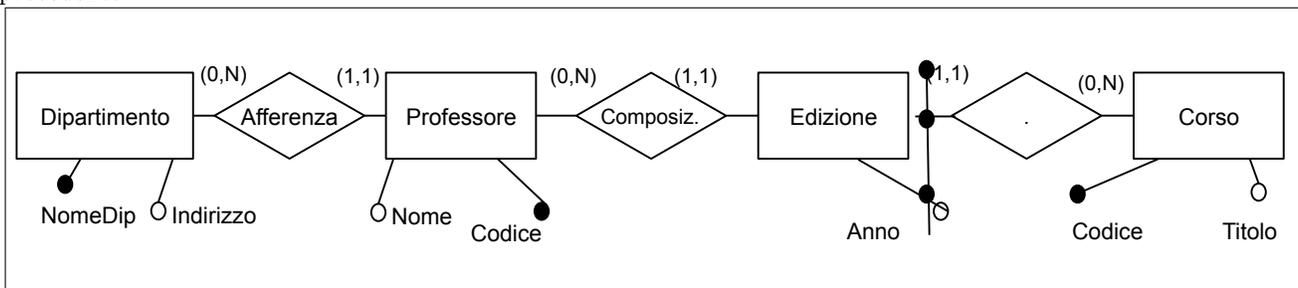
la chiave è CodiceCorso, Anno; dipendenze, oltre a quelle che hanno l'intera chiave a primo membro, sono le seguenti, che violano quindi la BCNF

- CodiceProf → NomeProf, Dipartimento
- Dipartimento → Indirizzo
- CodiceCorso → TitoloCorso

2. mostrare una buona decomposizione che soddisfi la BCNF (indicare gli schemi delle relazioni, con le relative chiavi)

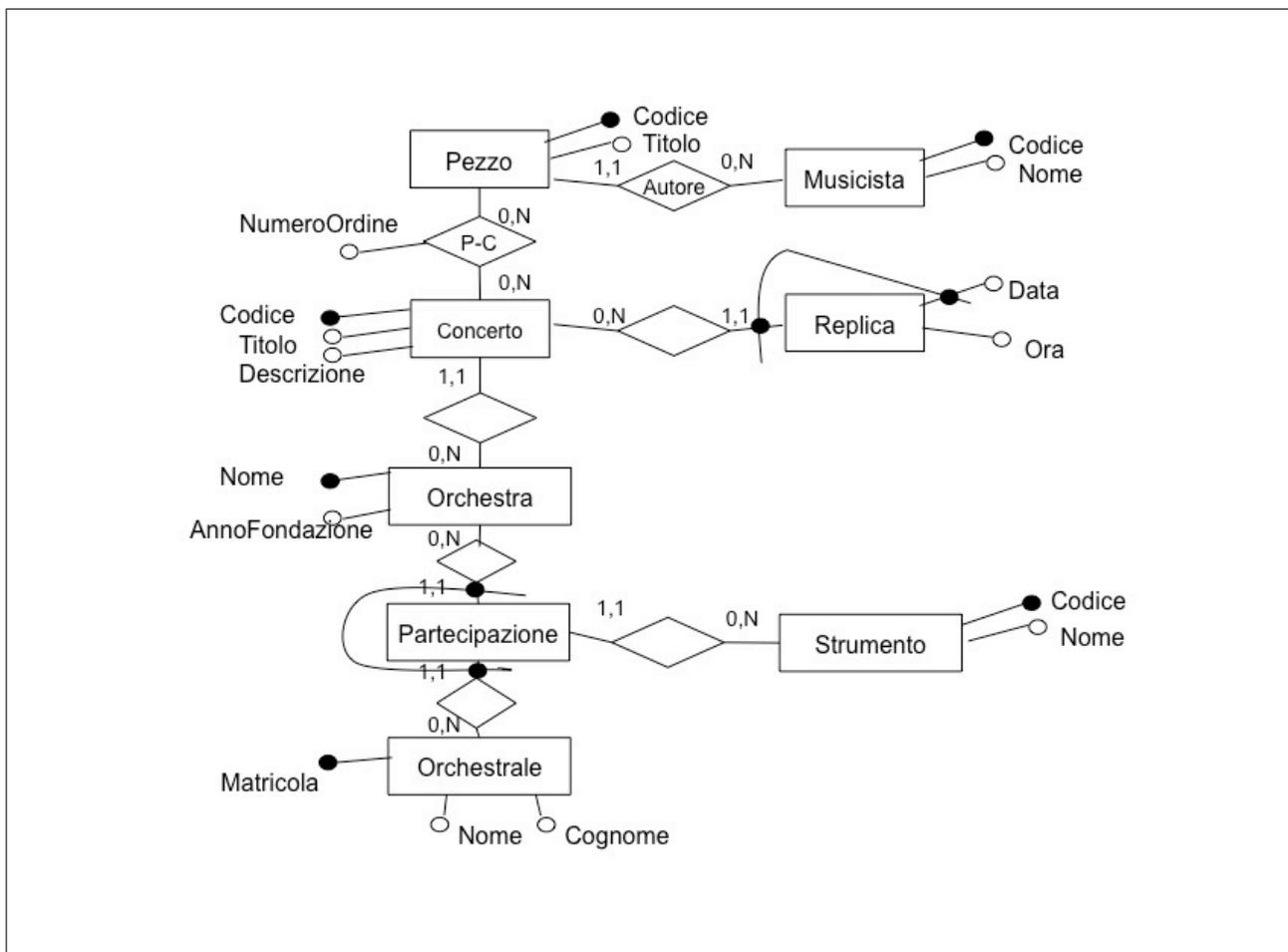
- R0 (CodiceCorso, Anno, CodiceProf)
- R1 (CodiceProf, NomeProf, Dipartimento)
- R2 (Dipartimento, Indirizzo)
- R3 (CodiceCorso, TitoloCorso)

**Domanda 2** (10%) Mostrare uno schema ER che descriva la realtà di interesse illustrata nella domanda precedente.



**Domanda 3** (30%) Mostrare lo schema concettuale per una base di dati per un programma di concerti, secondo le seguenti specifiche.

- Ogni concerto ha un codice, un titolo e una descrizione, ed è costituito da una sequenza di pezzi musicali diversi (interessa l'ordine)
- Ogni pezzo ha un codice, un titolo e un autore (con codice e nome); uno stesso pezzo può essere rappresentato in diversi concerti.
- Ogni concerto è eseguito da un'orchestra; ogni orchestra ha un nome (che la identifica), un anno di fondazione e un insieme di orchestrali.
- Ogni orchestrale ha una matricola (univoca nell'ambito della base di dati), nome e cognome, può partecipare a più orchestre, in ciascuna delle quali suona uno e un solo strumento, ma in orchestre diverse può suonare strumenti diversi.
- Ogni strumento ha un codice e un nome
- Ogni concerto è tenuto più volte, in giorni diversi e in orari che possono variare



**Domanda 4** (25%) Considerare la seguente base di dati relazionale:

- PRODOTTI(Codice, Descrizione, Marca)
- VENDITE(Prodotto, Mese, Anno, Quantità)  
con vincolo di integrità referenziale fra Prodotto e la relazione PRODOTTI

Con riferimento a tale base di dati (supponendo che, per i mesi senza vendite, i valori delle quantità siano pari a zero e quindi tutti i mesi di interesse siano riportati):

1. formulare in algebra relazionale l'interrogazione che trova codice, descrizione e marca per ogni prodotto che abbia almeno una vendita nell'anno 2013.

$$\pi_{\text{Codice, Descrizione, Marca}}(\text{PRODOTTI} \bowtie_{\text{Codice=Prodotto}} \sigma_{\text{Anno=2013} \wedge \text{Quantita} > 0}(\text{VENDITE}))$$

2. formulare in SQL la stessa interrogazione del punto precedente.

```
select distinct prodotti.*
from prodotti join vendite on codice = prodotto
where anno = 2013 and quantita > 0
```

3. formulare in SQL l'interrogazione che trova i prodotti che in un certo mese del 2014 hanno avuto vendite inferiori a quelle dello stesso mese del 2013; mostrare il codice e la descrizione del prodotto e le due quantità.

```
select codice,
       descrizione,
       v2013.quantita as quantita2013,
       v2014.quantita as quantita2014
from vendite v2013 join vendite v2014 on v2013.prodotto = v2014.prodotto
      join prodotti on v2013.prodotto = codice
where v2014.quantita < v2013.quantita
      and v2014.anno = 2014
      and v2013.anno = 2013
      and v2013.mese = v2014.mese
```

4. formulare in SQL l'interrogazione che trova le coppie di prodotti di una stessa marca; mostrare i due codici, le due descrizioni e la marca.

```
select p1.codice as codice1, p1.descrizione as desc1,
       p2.codice as codice2, p2.descrizione as desc2,
       p1.marca
from   prodotti as p1 join prodotti as p2
      on (p1.marca = p2.marca and p1.codice > p2.codice)
```

**Domanda 5** (20%) Considerare una relazione sullo schema:

- PARTITE(ID, Data, Squadra1, Squadra2, Reti1, Reti2)

e una vista definita su di essa, con la seguente istruzione SQL

```
CREATE VIEW partitebis AS
    SELECT data, squadra1 AS squadra, squadra2 AS avversaria, reti1 AS rfatte, reti2 AS rsubite
    FROM partite
UNION
    SELECT data, squadra2 AS squadra, squadra1 AS avversaria, reti2 AS rfatte, reti1 AS rsubite
    FROM partite
```

La figura seguente mostra parte di un'istanza della relazione PARTITE e della corrispondente vista PARTITEBIS.

PARTITE					
ID	Data	Squadra1	Squadra2	Reti1	Reti2
1	12/06	Brasile	Croazia	3	1
2	13/06	Messico	Camerun	1	0
3	13/06	Spagna	Olanda	1	5
...	...	...	...	...	...

PARTITEBIS				
Data	Squadra	Avversaria	RFatte	RSubite
12/06	Brasile	Croazia	3	1
12/06	Croazia	Brasile	1	3
13/06	Messico	Camerun	1	0
13/06	Camerun	Messico	0	1
13/06	Spagna	Olanda	1	5
13/06	Olanda	Spagna	5	1
...	...	...	...	...

1. Formulare in SQL l'interrogazione che genera, per ogni squadra che ha vinto almeno una partita, il numero di partite vinte (nota: l'indicazione di ignorare le squadre che non hanno vinto partite semplifica l'interrogazione; si consiglia di utilizzare la vista PARTITEBIS, che semplifica la scrittura dell'interrogazione)

```
select squadra, count(*)
from partitebis
where rfatte > rsubite
group by squadra
```

A titolo di curiosità, si segnala che l'interrogazione che include anche le squadre che non hanno vinto partite può essere formulata come segue:

```
select squadra, count(*) as vittorie
from partitebis
where rfatte > rsubite
group by squadra
union
select distinct squadra, 0
from partitebis p
where not exists (
    select *
    from partitebis
    where squadra = p.squadra
    and rfatte > rsubite )
```

2. Si vuole produrre la “classifica” indicando, per ogni squadra il numero di partite vinte, pareggiate e perse (sempre con riferimento alle squadre che hanno almeno una partita vinta, una pareggiata e una persa). Descrivere la struttura dell’interrogazione SQL da utilizzare allo scopo, anche senza scrivere tutto il codice.

Si possono usare tre viste, basate su interrogazioni come la precedente e poi il join delle tre viste

```
create view contavittorie as
  select squadra, count(*) as vittorie
  from partitebis
  where rfatte > rsubite
  group by squadra

create view contapareggi as
  select squadra, count(*) as pareggi
  from partitebis
  where rfatte = rsubite
  group by squadra

create view contasconfitte as
  select squadra, count(*) as sconfitte
  from partitebis
  where rfatte < rsubite
  group by squadra

select v.squadra, vittorie, pareggi, sconfitte
  from contavittorie v, contapareggi p, contasconfitte s
  where v.squadra = p.squadra and p.squadra = s.squadra
```