

## PROVA COMPLETA - Basi di Dati e Lab

09/01/2020

Corso di Laurea in Ingegneria Informatica

Prof. Sonia Bergamaschi

### Esercizio 1 (punti 5)

*È necessario totalizzare almeno 3 punti in questo esercizio affinché il resto della prova venga valutato.*

Illustrare il concetto di reificazione in E/R, indicando i vari casi e gli identificatori corrispondenti.

### Esercizio 2 (punti 11)

Si vuole realizzare un sistema informativo per la memorizzazione dei dati di un'azienda di scommesse sportive online di partite di calcio.

L'azienda possiede diverse agenzie di scommesse, ciascuna delle quali è identificata dalla città in cui si trova e da un numero progressivo. L'azienda ha diversi dipendenti (per i quali vengono riportati nome, cognome, codice fiscale e indirizzo) che possono lavorare per una o più agenzie. Ogni agenzia ha un responsabile scelto tra i dipendenti. Un dipendente può essere responsabile di una sola agenzia.

È possibile scommettere sul risultato e/o sugli eventi di una partita di calcio. Per ogni partita di calcio vengono indicati i nomi delle due squadre, la data e l'ora in cui sarà o è stata disputata la partita. Nel caso la partita sia già stata disputata viene riportato anche il tabellino che comprende il risultato e i dati su ogni goal (minuto e marcatore). Ci sono 3 tipi di eventi su cui si può scommettere, a ciascuno dei quali è associato una quota: risultato (primo tempo/finale), goal/no goal, under/over N goal (N è il numero di goal che può essere 1.5 o 2.5). Un cliente, identificato dagli usuali dati anagrafici, può recarsi in una delle agenzie per effettuare una giocata sugli eventi che accadono in una o più partite. Ogni giocata è identificata da un numero progressivo globale, l'importo scommesso e la quota complessiva. Una giocata deve essere etichettata come chiusa se tutte le partite di cui è composta sono state disputate. In tal caso viene indicato anche se la giocata è vincente o perdente.

L'azienda possiede diversi ispettori che verificano il corretto funzionamento delle operazioni nelle varie agenzie. Ciascuna agenzia può ricevere ogni anno da 3 a 5 ispezioni. L'ispettore fornisce un report della sua visita nelle quali vengono riportate la data dell'ispezione, una breve descrizione e l'esito che può essere "positivo", "negativo" o "da verificare". Le agenzie che hanno ricevuto un esito "negativo" non possono accettare scommesse fintanto non viene effettuata una nuova visita con esito "positivo", mentre quelle "da verificare" devono ricevere una nuova visita da un ispettore entro un mese.

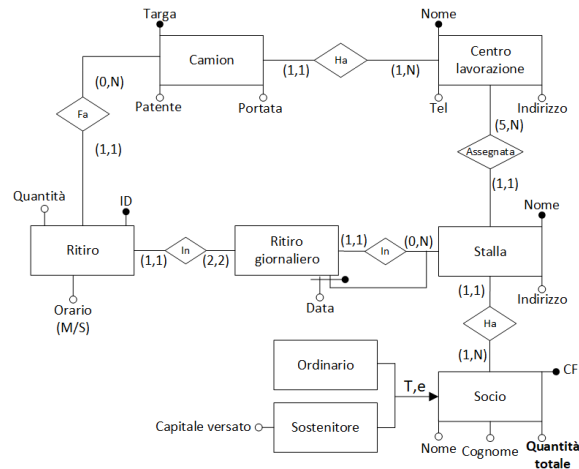
### Viene richiesto di:

- a) Progettare lo schema E/R, evidenziando gli eventuali vincoli non esprimibili nel modello. (9 punti)
- b) Modificare lo schema precedente aggiungendo il seguente vincolo: (2 punti)

Ciascun cliente può effettuare al massimo 5 giocate al giorno (considerando tutte le agenzie).

### Esercizio 3 (punti 6)

Sia dato il seguente schema E/R



A partire dallo schema, dalle informazioni contenute nella tabella dei volumi dei dati e dalla frequenza delle operazioni, viene richiesto di:

1. Determinare se conviene o meno mantenere l'attributo derivato "quantità totale" (\*), considerando le seguenti operazioni (3 punti):

**Operazione 1:** dato il codice fiscale di un socio, ottenere la quantità totale del latte ritirato dalle sue stalle.

**Operazione 2:** inserimento di un nuovo ritiro, si suppongano noti e validi i dati del camion e del ritiro giornaliero.

**Frequenza:** operazione 1: 100/giorno, operazione 2: 400/giorno.

\* con quantità totale si intende la somma delle quantità dei ritiri effettuati dalle stalle di proprietà del socio.

2. Produrre lo schema logico relazionale in terza forma normale evidenziando eventuali vincoli non esprimibili. (3 punti)

**Tabella dei volumi dei dati**

CONCETTO	TIPO	VOL.
Centro lavorazione	E	10
Camion	E	20
Ritiro	E	1200
Stalla	E	200
Socio	E	80

*Il 20% sono soci sostenitori, l'80% sono ordinari*

#### Esercizio 4 (punti 11)

Sia dato il seguente schema relazionale relativo alla gestione degli acquisti di biglietti per voli di una compagnia aerea.

Utente (username, Nome, Cognome, Cittadinanza)

Aeroporto (CodA, Città, Stato)

Volo (CodV, Partenza, Destinazione, Data)

**FK** Partenza **REFERENCES** Aeroporto (\*)

**FK** Destinazione **REFERENCES** Aeroporto (\*)

Acquisto (username, CodV, prezzo)

**FK** username **REFERENCES** Utente

**FK** CodV **REFERENCES** Volo

(\*) *Si noti che Partenza e Destinazione fanno riferimento al codice aeroporto CodA e non al nome della città dell'aeroporto.*

Si richiede di scrivere in algebra relazionale e in SQL le seguenti interrogazioni:

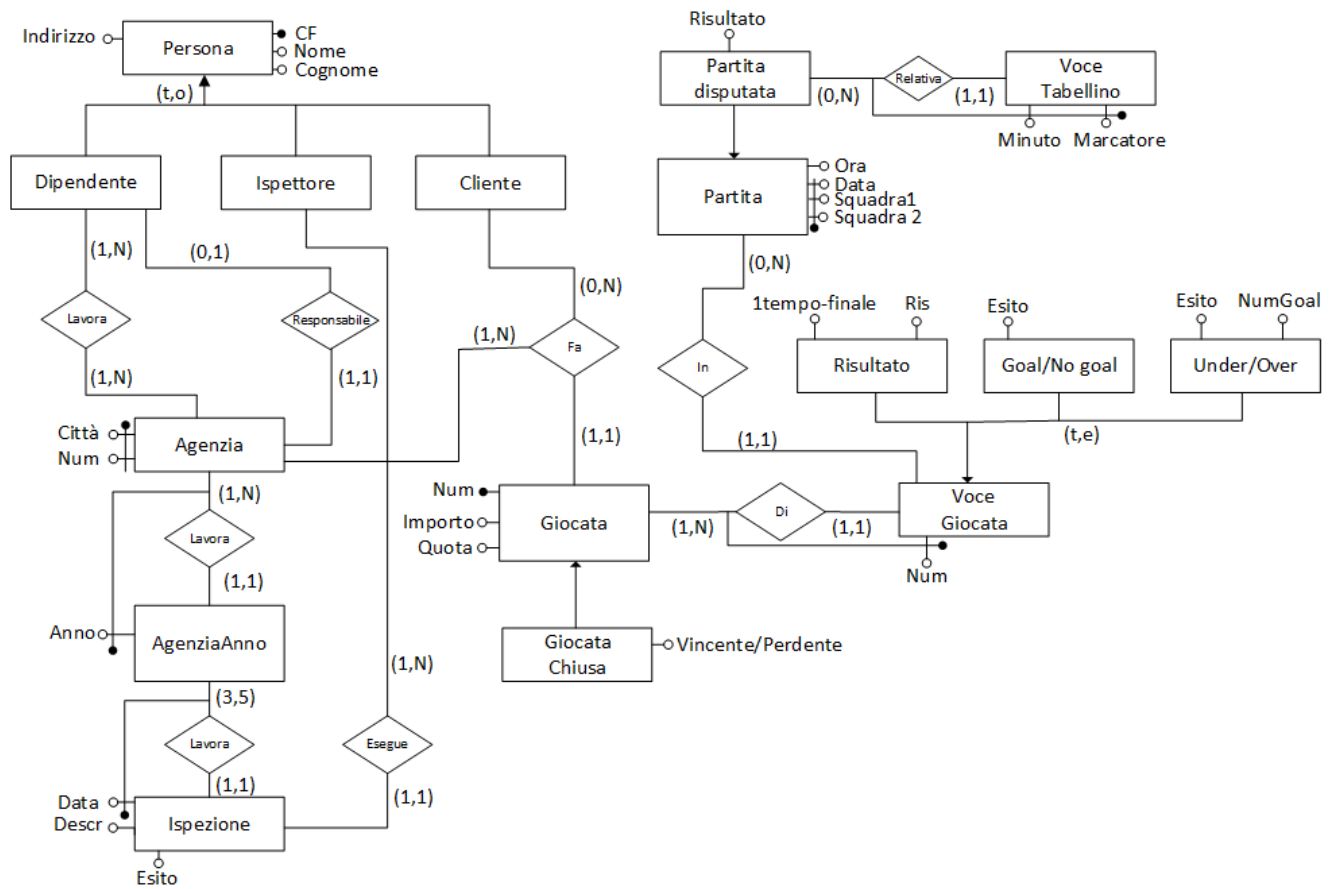
- a) Selezionare gli utenti che nel 2014 non hanno acquistato voli con destinazione “New York”.  
(punti 2+2)
- b) Selezionare gli utenti che nel maggio 2010 sono partiti da tutti gli aeroporti di Milano. (punti 2+2)

Si richiede di scrivere in SQL la seguente interrogazione (può essere risolta anche con più viste):

- c) Creare una vista che mostri per ogni aeroporto, il numero di voli partiti, il numero di voli arrivati, il numero distinto di utenti arrivati, il numero distinto di utenti partiti. Mostrare solamente gli aeroporti che sono stati visitati (partenze + arrivi) da almeno 10000 persone diverse. (punti 3)

## SOLUZIONE

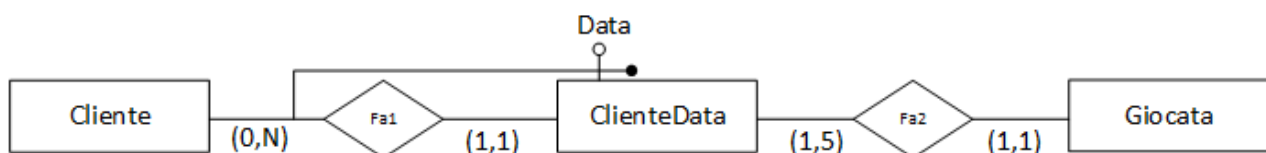
### Esercizio 2



### Vincoli non esprimibili:

- “Una giocata deve essere etichettata come chiusa se tutte le partite di cui è composta sono state disputate.”
- “Le agenzie che hanno ricevuto un esito “negativo” non possono accettare giocate fintanto non viene effettuato una nuova visita con esito “positivo”, mentre quelle “da verificare” devono ricevere una nuova visita da un ispettore entro un mese.”

### Vincolo aggiuntivo



### Esercizio 3

#### Operazione 1 con dato derivato

CONCETTO	ACCESSI	TIPO	DESCRIZIONE / MOTIVAZIONE DELL'OPERAZIONE
Socio	1	L	Leggo quantità totale

#### Operazione 1 senza dato derivato

CONCETTO	ACCESSI	TIPO	DESCRIZIONE / MOTIVAZIONE DELL'OPERAZIONE
Ha	2,5	L	Leggo stalle associate al socio. $200 \text{ stalle}, 80 \text{ soci}, 200/80 = 2,5 \text{ stalle per socio.}$
Stalla	2,5	L	
In	7,5	L	Leggo ritiri giornalieri. Ogni ritiro è associato a 2 ritiri giornalieri, quindi $1200/2 = 600$ . 600 istanze di ritiro giornaliero, 200 stalle. $600/200 = 3$ ritiri per stalla. $3 * 2,5 = 7,5$
Ritiro Giornaliero	7,5	L	
In	15	L	600 ritiri, 1200 ritiri giornalieri, $1200/600 = 2$ ritiri per ogni giorno.
Ritiro	15	L	

#### Operazione 2 con dato derivato

CONCETTO	ACCESSI	TIPO	DESCRIZIONE / MOTIVAZIONE DELL'OPERAZIONE
Ritiro	1	S	Inserisco il ritiro
In	1	S	Associo il ritiro al ritiro giornaliero
Fa	1	S	Inserisco il camion
Ha	1	L	Avendo i dati del ritiro giornaliero ho il nome della stalla, leggo il CF del socio.
Socio	1	L	Leggo valore attuale
Socio	1	S	Scrivo nuovo valore

#### Operazione 2 senza dato derivato

CONCETTO	ACCESSI	TIPO	DESCRIZIONE / MOTIVAZIONE DELL'OPERAZIONE
Ritiro	1	S	Inserisco ritiro
In	1	S	Associo il ritiro al ritiro giornaliero
Fa	1	S	Associo il camion

**OP1 con dato derivato:** costo 1

**OP1 senza dato derivato:** costo 50

**OP2 con dato derivato:** costo 10

**OP2 senza dato derivato:** costo 6

**Complessivo senza dato:**  $50 \cdot 100 + 6 \cdot 400 = 7400$  accessi

**Complessivo con dato:**  $1 \cdot 100 + 10 \cdot 400 = 4100$  accessi

**CONVIENE MANTENERE IL DATO DERIVATO**

### Schema logico

Centro\_lavorazione (Nome, Indirizzo, Tel)

Camion (Targa, Patente, Portata, Nome)

**FK:** Nome **REFERENCES** Centro\_lavorazione **NOT NULL**

Socio (CF, Nome, Cognome, Quantità totale, Sel, Capitale versato)

**Dominio** Sel {ordinario, sostenitore}

(\*)

Stalla (Nome, Indirizzo, CF, NomeCentro)

**FK:** CF **REFERENCES** Socio **NOT NULL**

**FK:** NomeCentro **REFERENCES** Centro\_lavorazione **NOT NULL**

(\*\*)

Ritiro (ID, Quantità, Orario, Nome, Data, Targa)

**FK:** Nome **REFERENCES** Stalla **NOT NULL**

**FK:** Targa **REFERENCES** Camion **NOT NULL**

(\*\*\*)

(\*) Collasso verso l'alto

(\*\*) Vincolo non esprimibile: minimo 5 stalle per centro di lavorazione

(\*\*\*) Vincolo non esprimibile: massimo 2 ritiri al giorno per stalla

(\*\*\*\*) Si noti che è stata eliminata l'entità "ritiro\_giornaliero"

### Esercizio 4

- a) Selezionare gli utenti che nel 2014 non hanno acquistato voli con destinazione "New York".  
(punti 2+2)

Seleziona i voli del 2014

$$S1 = \sigma_{data \geq '1-1-2014' \wedge data \leq '31-12-2014'}(Volo)$$

Seleziona gli aeroporti di New York

$$S2 = \sigma_{città='New York'}(Aeroporto)$$

Seleziona i voli del 2014 con destinazione New York

$$S3 = S1 \bowtie_{Destinazione=CodA} (S2)$$

Seleziona gli utenti che nel 2014 non hanno acquistato voli con destinazione New York

$$\pi_{username}(Utente) - \pi_{username}(Utente \bowtie Acquisto \bowtie S3)$$

```

SELECT *
FROM Utente U
WHERE username NOT IN (
    SELECT username
    FROM Acquisto A
    JOIN Volo V ON V.CodV = A.CodV
    JOIN Aeroporto AE ON AE.CodA = V.Destinazione
    WHERE YEAR(V.Data) = 2014
    AND AE.Città = 'New York'
)

```

Selezionare gli utenti che nel maggio 2010 sono partiti da tutti gli aeroporti di Milano. (punti 2+2)

$$S1 = \pi_{CodA}(\sigma_{città='Milano'}(Aeroporto))$$

$$S2 = \pi_{username,partenza}(\sigma_{data \geq '1-5-2010' \wedge data \leq '31-5-2010'}(Volo) \bowtie Acquisto)$$

$$Utente \bowtie (S2 \div S1)$$

Selezionare gli utenti per cui NON ESISTE un aeroporto di Milano da cui NON sono partiti nel mese di maggio 2010

```

SELECT *
FROM Utente U
WHERE NOT EXISTS (
    SELECT *
    FROM Aeroporto AE
    WHERE AE.città = 'Milano'
    AND NOT EXISTS (
        SELECT *
        FROM Acquisto A
        JOIN Volo V ON A.CodV = V.CodV
        WHERE V.Partenza = AE.CodA
        AND A.username = U.username
        AND V.Data >= '1-5-2010' AND V.Data <= '31-5-2010'
    )
)

```

- b) Creare una vista che mostri per ogni aeroporto, il numero di voli partiti, il numero di voli arrivati, il numero distinto di utenti arrivati, il numero distinto di utenti partiti. Mostrare solamente gli aeroporti che sono stati visitati (partenze + arrivi) da almeno 10000 persone diverse. **(punti 3)**

```
CREATE VIEW v1 AS
  SELECT A.CodA, COUNT(DISTINCT VP.codV) AS num_voli_partiti,
    COUNT(DISTINCT VA.codV) AS num_voli_arrivati,
    COUNT(DISTINCT AP.username) AS utenti_partiti,
    COUNT(DISTINCT AA.username) AS utenti_arrivati
FROM aeroporto A
JOIN VOLO VP ON VP.Partenza = A.CodA
JOIN VOLO VA ON VA.Destinazione = A.CodA
JOIN acquisto AP ON AP.CodV = VP.CodV
JOIN acquisto AA ON AA.CodV = VA.CodV
GROUP BY A.CodA
HAVING (COUNT(DISTINCT AP.username) +
    COUNT(DISTINCT AA.username)) > 10000
```