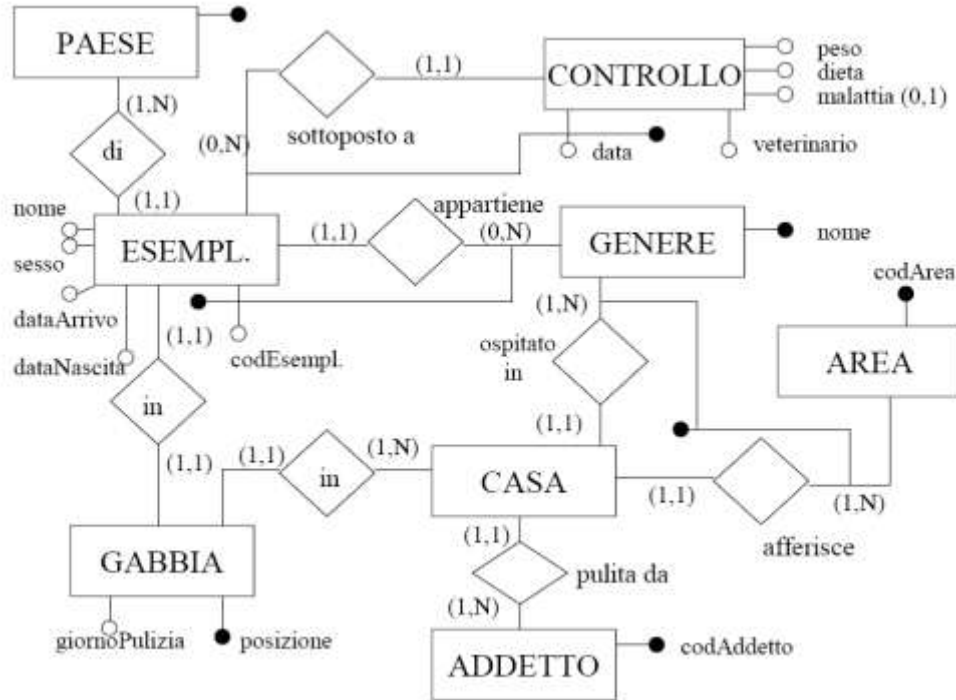


Esercizio 1

Dato il seguente schema E/R operativo si individui un *fatto* di interesse si definisca la struttura relazionale dello *snowflake* relativo, mostrando i passi dell'*attribute tree* e del *fact scheme*.



Svolgimento

Schema relazionale:

PAESE(nomePaese)

ESEMPLARE(codEsemplare, genere:GENERE, nome, sesso, dataArrivo, dataNascita, gabbia:GABBIA, paese:PAESE)

GABBIA(posizione, giornoPulizia, area:CASA, genere:CASA)

CASA(codArea:AREA, genere:GENERE, addetto:ADDETTO)

ADDETTO(codAddetto)

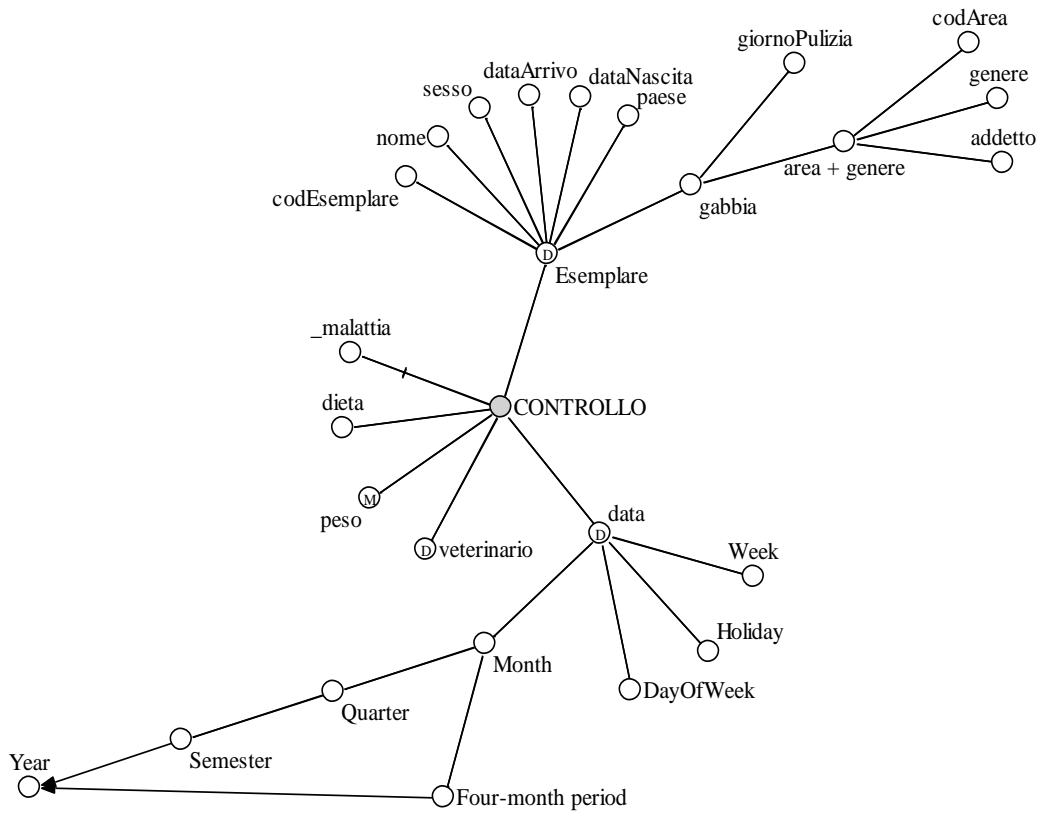
AREA(codArea)

GENERE(nome)

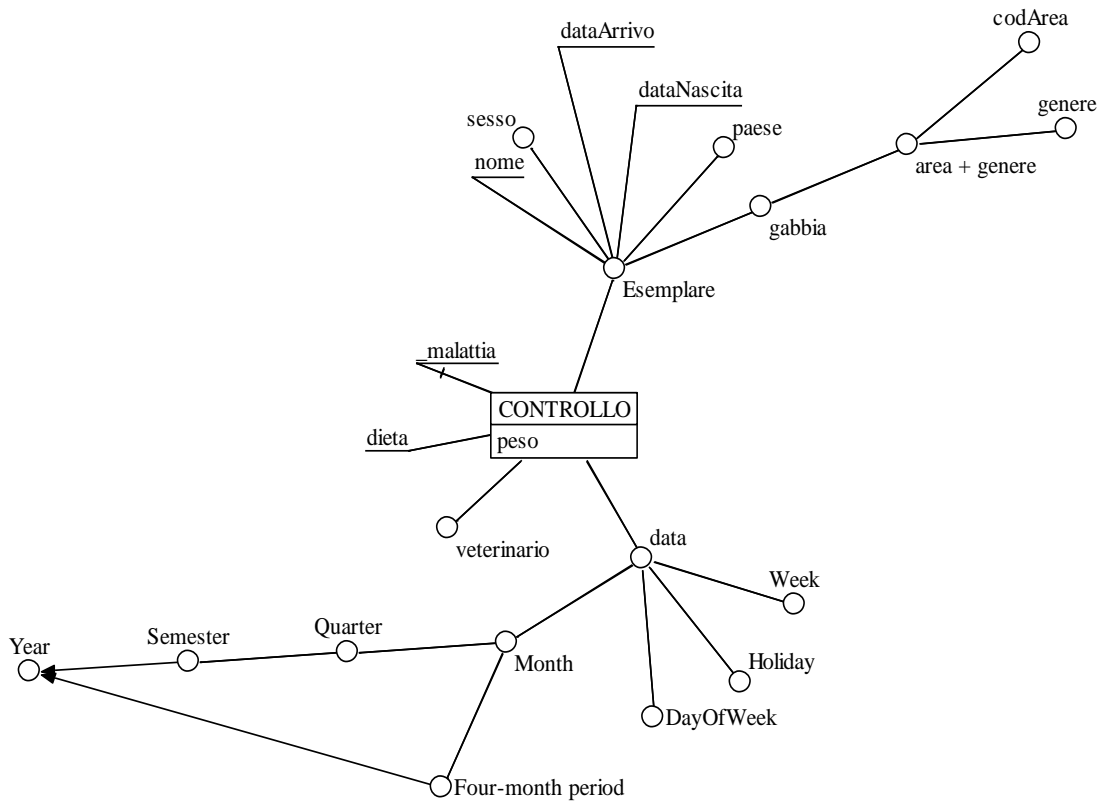
CONTROLLO(data, codEsemplare:ESEMPLARE, genere:ESEMPLARE, veterinario, peso, dieta, malattia*)

Fatto scelto: Controllo

Attribute tree:



Fact-Schema.



Esercizio 2

E' data la query:

```
SELECT COMPAGNIA.nome, VOLO.partenza, VOLO.destinazione
FROM COMPAGNIA, VOLO
WHERE COMPAGNIA.cod = VOLO.codCompagnia
AND VOLO.partenza = 'Roma'
AND COMPAGNIA.nazione = 'Olanda'
```

sulle relazioni:

COMPAGNIA(cod, nome, aeroporto_principale, nazione)
VOLO(numVolo, codCompagnia:COMPAGNIA, partenza, destinazione)

Nell'ipotesi che:

$NP_{COM} = 160$ $NT_{COM} = 500$
 $NP_{VOL} = 550$ $NT_{VOL} = 150000$
 $NK_{Partenza} = 400$ $NK_{nazione} = 40$

e sapendo che su COMPAGNIA.cod è costruito un indice clustered con $NL_{cod} = 10$ e $h = 2$

- 1) Si calcoli il numero di *tuple attese* della query
- 2) Si determini il miglior *piano d'accesso* per la risoluzione della query fra i due seguenti:
 1. nested-loop con relazione interna VOLO
 2. nested-loop con relazione interna COMPAGNIA

Svolgimento

$f(\text{nazione}) = 1/NK_{\text{nazione}} = 1/40$
 $f(\text{partenza}) = 1/NK_{\text{partenza}} = 1/400$

1) *tuple attese*

$$ET : \lceil 1/40 * 1/400 * 150000 \rceil = 10$$

2) *piani d'accesso*

1. nested-loop con relazione interna VOLO

$$C_{TOT} = C_a(COM) + ET_{COM} * C_a(VOL)$$

$$C_a(COM) = NP_{COM} = 160 \text{ (seq.)}$$

$$C_a(VOL) = NP_{VOL} = 550 \text{ (seq.)}$$

$$ET_{COM} = \lceil NT_{COM} * f(\text{nazione}) \rceil = \lceil 500/40 \rceil = 13$$

$$C_{TOT} = 160 + 13 * 550 = 7310$$

2. nested-loop con relazione interna COMPAGNIA

$$C_{TOT} = C_a(VOL) + ET_{VOL} * C_a(COM)$$

$$C_a(VOL) = NP_{VOL} = 550 \text{ (seq.)}$$

$$a) C_a(COM) = NP_{COM} = 160 \text{ (seq.)}$$

$$b) C_a(COM) = h-1 + \lceil NL_{cod} * f(\text{cod=val}) \rceil + \lceil NP_{COM} * f(\text{cod=val}) \rceil =$$
$$1 + \lceil 10/500 \rceil + \lceil 160/500 \rceil = 3 \text{ (idx.)}$$

$$ET(VOL) = \lceil NT_{VOL} * f(\text{partenza}) \rceil = \lceil 150000/400 \rceil = 375$$

$$C_{TOT} = 550 + 375 * 3 = 1675$$

Esercizio 3

Dato il seguente schema relazionale:

SOCIO (tesserino, nome, indirizzo, città, età, telefono, dataIscrizione)

CORSO(codCorso, nome, descrizione, dataInizio, numeroIscritti)

ISCRIZIONE (tesserino: SOCIO, codCorso: CORSO)

1. Scrivere la *stored procedure* che visualizza i corsi (nome, descrizione) che non hanno nessun iscritto.
2. Creare un *trigger* che tiene aggiornato il dato derivato "numeroIscritti". A fronte dell'inserimento di un nuovo iscritto incrementa il campo numeroIscritti del relativo corso.

Svolgimento

1. CREATE PROCEDURE CorsiVuoti
AS
SELECT nome, descrizione FROM CORSO
WHERE numeroIscritti=0
2. CREATE TRIGGER AggiornaIscritti
ON Iscrizione
FOR INSERT
AS
UPDATE CORSO
SET numeroIscritti= numeroIscritti+1
WHERE CORSO.codCorso IN (SELECT codCorso FROM INSERTED)

Esercizio 4

Un B-tree di ordine $g=2$ (senza gestione dell'overflow) è costruito inserendo, in successione, le chiavi 11, 12, 13, 15, 100, 60, 39,38, 70, 72, 47, 81, 96, 1, 5, 6. Disegnare l'albero al termine dell'inserimento.

Svolgimento

La capacità delle pagine è di 4 record.

