

Esercizi Basi di Dati II: Bozza 1.0

Adriano Peron

6 Novembre 2015

1 Indici

Esercizio 1 *Si consideri il seguente insieme di punti $A(6, 2)$, $B(6, 4)$, $C(4, 4)$, $D(8, 8)$, $E(4, 6)$, $F(1, 6)$, $G(2, 8)$, $H(1, 2)$ collocati in una superficie 10×10 . Si costruisca un G -albero di grado 3 per la memorizzazione dei punti. Una volta costruito l'albero si operi l'inserimento del punto $K(9, 1)$. Si supponga inoltre di dover ricercare i punti compresi nell'intervallo di estremi $(3, 3)$ e $(8, 7)$: si indichi come l'indice a G -albero viene utilizzato per tale ricerca.*

Esercizio 2 (punti 6) *Si consideri un $B+$ albero di ordine 3 (sia per i nodi sia per le foglie). Si costruisca l'albero per la seguente sequenza di inserimenti: 30, 25, 27, 45, 48, 26, 28, 46, 47, 50. Si esegua infine la seguente sequenza di cancellazioni: 45, 26, 27, 28.*

Esercizio 3 *Si indichino brevemente la differenze tra gli indici a $B+$ albero e quelli a R -albero indicando per quali tipologie di interrogazioni sono adatti.*

Esercizio 4 *Si descrivano in modo sintetico similitudini e differenze tra i $B+$ alberi e gli R -alberi. Si consideri poi un R -albero con nodi di capienza massima 2 e di capienza minima 1 e si costruisca l' R -albero per l'inserimento dei seguenti rettangoli indicati dalle coppie di vertici left-bottom, right-top: $A (1, 1)(2, 2)$; $B (2, 8)(5, 10)$; $C (4, 7)(6, 9)$; $D (1, 1)(4, 4)$; $E (7, 8)(10, 9)$; $F (1, 6)(2, 7)$. Si dica come viene usato l'albero (indice) per recuperare gli elementi intersecanti l'area rettangolare di ricerca $(1, 3)(3, 10)$.*

Esercizio 5 *Si descrivano gli indici a liste invertite. Si analizzi il costo della ricerca per intervallo nel caso in cui l'attributo indicizzato non sia di ordinamento.*

2 Ottimizzazione di query

Esercizio 6 *Si considerino le tabelle di schema $Studenti(Matricola, AnnoImm, Nome, Cognome)$ e $Esami(Matricola, Insegnamento, Data, Voto)$, con $Matricola$ chiave per studenti, e l'interrogazione*

*SELECT **
FROM Studenti S, Esami E
WHERE S.matricola = E.matricola AND AnnoImm=2000 AND Voto=30
 Si supponga che siano disponibili le seguenti informazioni: $N_{reg}(Studenti) = 10000$, $N_{pag}(Studenti) = 1000$, $N_{reg}(Esami) = 150000$, $N_{pag}(Esami) = 7500$. Si supponga che l'unico indice disponibile sia sull'attributo Matricola di Studenti.

1. Si fornisca l'albero trasformato degli operatori logici corrispondenti alla *SELECT* (anticipo di selezioni e proiezioni sul prodotto);
2. Si fornisca una stima delle dimensioni del risultato avvalendosi dell'ipotesi di equidistribuzione (gli anni vanno dal 1998 al 2004);
3. Si consideri la possibilità che l'indice su Matricola sia un indice hash o un indice B^+ (con $N_{Leaf}(Matricola) = 100$). Si consideri quale indice risulta più vantaggioso per un *index nested loop*.

Esercizio 7 Si consideri l'interrogazione

*SELECT **
FROM R1, R2
WHERE R1.C = R2.C AND
R1.A = 77 AND R2.B < 500

Si supponga che $N_{reg}(R1) = 10000$, $N_{reg}(R2) = 500$, $N_{pag}(R1) = 1000$, $N_{pag}(R2) = 100$ e che esistano i seguenti indici:
 di ordinamento su $R2.C$ con $N_{key}(R2.C) = 500$ e $N_{leaf}(R2.C) = 10$;
 indice non di ordinamento su $R1.C$ con $N_{key}(R1.C) = 500$ e $N_{leaf}(R1.C) = 100$;
 indice di ordinamento su $R1.A$ con $N_{key}(R1.A) = 1000$ e $N_{leaf}(R1.A) = 100$.
 Si stimi la dimensione del risultato e si indichi il costo della strategia ottima per il metodo *index nested loop*.

Esercizio 8 Si considerino le tabelle

Docente(Codice, Dipartimento, AreaRicerca, Stipendi),
Dipartimenti(Cod - Dip, Nome, Tel, Citta)

e l'interrogazione

SELECT Codice
FROM Docente, Dipartimenti
WHERE Dipartimento=Cod-Dip AND
AreaRicerca= 'Informatica' AND Citta = 'Napoli' AND
Stipendio > = 2000.

Si supponga sulla tabella *Docente* vi siano indici B^+ su *Codice*, B^+ di ordinamento su *Dipartimento*, hash su *Stipendio*.

Si supponga sulla tabella *Dipartimenti* vi siano indici B^+ su *Cod - Dip* e B^+ di ordinamento su *Citta*. Si supponga inoltre di disporre delle seguenti informazioni statistiche: $N_{key}(AreaRicerca) = 100$, $Min(Stipendio) = 1000$,

$Max(Stipendio) = 4000$, $N_{reg}(Docente) = 10000$ $N_{pag}(Docente) = 110$,
 $N_{leaf}(Dipartimento) = 5$;
 $N_{key}(Citta) = 20$, $N_{reg}(Dipartimento) = 300$, $N_{pag}(Dipartimento) = 18$,
 $N_{leaf}(Citta) = 3$.

Considerando l'Index Nested Loop si indichi la strategia che l'ottimizzatore dovrebbe adottare per l'esecuzione dell'interrogazione (argomentando la scelta con i dati statistici forniti). Si dia, inoltre, una stima della dimensione del risultato.

Esercizio 9 Si consideri la tabella

$STUDENTI(Matricola, Nome, Cognome, Sesso,$
 $DataNascita, CittaNascita, CittaResidenza)$

e l'interrogazione

$SELECT$ $Matricola, Nome, Cognome$
 $FROM$ $STUDENTI$

$WHERE ((Sesso = 'M' AND CittaNascita = 'Napoli') OR (DataNascita >= '1985-01-01' AND Sesso = 'F')) AND CittaResidenza = Napoli$

Si supponga di avere un indice bitmap su Sesso, e indici a liste invertite su CittaNascita e CittaResidenza.

1. Scrivere l'albero logico dell'interrogazione;
2. Si dia una stima della dimensione del risultato;
3. Si discutano le possibili strategie per l'esecuzione dell'interrogazione stimandone i costi.

Esercizio 10 Si considerino le tabelle

$Conducente(CF, Nome, Cognome, DataN)$,

$Autobus(Mat, Tipo, Alimentazione)$

$Servizio(CF, Matr, Data, Linea)$

e l'interrogazione

$SELECT$ $Nome, Cognome$

$FROM$ $Conducente$ $NATURAL JOIN$ $Autobus$ $NATURAL JOIN$ $Servizio$

$WHERE (Tipo = 'IVECO' AND Linea = 'C15') OR (alimentazione = 'Gas' AND Linea = 'C15')$.

Si scriva l'albero sintattico della interrogazione. Considerando l'Index Nested Loop come strategia di riferimento per implementare la giunzione, si introducano degli indici in maniera ragionata e si indichi la strategia che ragionevolmente l'ottimizzatore dovrebbe adottare per l'esecuzione dell'interrogazione argomentando le scelte fatte. Si dia per via teorica, inoltre, una stima della dimensione del risultato e per il piano di esecuzione scelto una stima teorica del costo.

Esercizio 11 Si considerino le tabelle

$Conducente(CF, Nome, Cognome, DataN)$,

$Autobus(Mat, Tipo, Alimentazione)$

Servizio(*CF, Matr, Data, Linea*)

e l'interrogazione

SELECT Nome, Cognome

FROM Conducente NATURAL JOIN Autobus NATURAL JOIN Servizio

WHERE (Tipo='IVECO' OR alimentazione = 'Gas') AND Data \geq 'DATE2008-06-01' AND Linea = 'C15'.

Si scriva l'albero sintattico della interrogazione. Considerando l'Index Nested Loop come strategia di riferimento per implementare la giunzione, si introducano degli indici in maniera ragionata e si indichi la strategia che ragionevolmente l'ottimizzatore dovrebbe adottare per l'esecuzione dell'interrogazione argomentando le scelte fatte. Si dia per via teorica, inoltre, una stima della dimensione del risultato e per il piano di esecuzione scelto una stima teorica del costo.

Esercizio 12 (7 punti) Si considerino la tabella

Docente(Codice, Dipartimento, SSD, Facolta, Citta), e l'interrogazione

SELECT Codice

FROM Docente

WHERE SSD='INF/01' OR (Facolta = 'Ingegneria' AND SSD='ING-INF/05')

Si supponga sulla tabella Docente vi siano indici B^+ (non di ordinamento) su SSD e di ordinamento su Codice.

Si considerino i seguenti dati: $N_{reg}(Docente) = 10000$ $N_{pag}(Docente) = 500$, $N_{leaf}(Docente) = 80$, $N_{key}(SSD) = 100$, $N_{leaf}(SSD) = 80$;

Si dia una stima della dimensione del risultato. Si dia una stima del costo dell'interrogazione spiegando in modo dettagliato l'algoritmo usato per la selezione (si indichi se vi sono e quali sono i predicati di ricerca, se vi sono e quali sono i predicati residui, come vengono usati etc).

Esercizio 13 Si considerino la tabella

Docente(Codice, Dipartimento, SSD, Facolta, Citta), e l'interrogazione

SELECT Codice

FROM Docente

WHERE SSD='INF/01' OR (Facolta = 'Ingegneria' AND SSD='ING-INF/05')

Si supponga sulla tabella Docente vi siano indici B^+ (non di ordinamento) su SSD e di ordinamento su Codice.

Si considerino i seguenti dati: $N_{reg}(Docente) = 10000$ $N_{pag}(Docente) = 500$, $N_{leaf}(Docente) = 80$, $N_{key}(SSD) = 100$, $N_{leaf}(SSD) = 80$;

Si dia una stima della dimensione del risultato. Si dia una stima del costo dell'interrogazione spiegando in modo dettagliato l'algoritmo usato per la selezione (si indichi se vi sono e quali sono i predicati di ricerca, se vi sono e quali sono i predicati residui, come vengono usati etc).

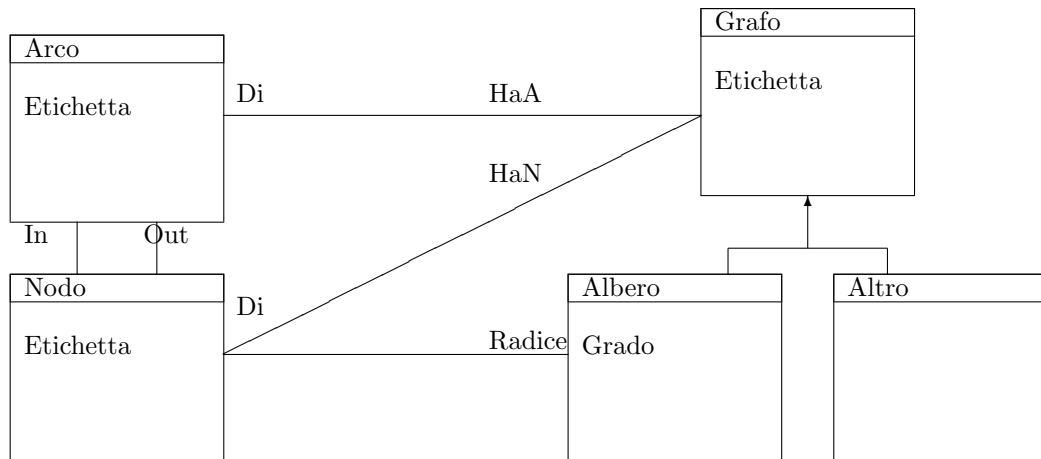


Figure 1: Figura 1

3 Data Base a Oggetti

Esercizio 14 Si consideri la bozza di Class Diagram in figura 1 che descrive la composizione di grafi. Ogni grafo ha un insieme di archi e di nodi. Gli archi e i nodi sono condivisi dai grafi. Un albero è la specializzazione di un grafo. Le etichette di grafi e nodi sono numeri interi.

Con riferimento a OQL si definiscano gli EXTENT dei tipi (la definizione completa dei tipi non è necessaria). Si scriva una interrogazione che per ogni grafo (indicato dalla sua etichetta) restituisca l'insieme delle etichette degli alberi che sono sottografi del grafo considerato (un sottografo ha l'insieme dei nodi e degli archi inclusi in quelli del grafo contenente).

Esercizio 15 Il Class diagram in figura 2 rappresenta la descrizione di Class diagrams. La classe CLASS descrive le classi. L'associazione ricorsiva descrive la relazione di specializzazione delle classi (Sub è la specializzazione della classe Sup). La cardinalità dell'associazione è Uno-a-Molti (una classe molte specializzazioni). La classe METODI indica quali metodi vengono associati ad una classe: Nome è il nome del metodo, TipoOut il tipo di dato restituito dal metodo. La cardinalità dell'associazione è Uno-a-Molti (una classe molti metodi, i metodi non sono condivisi dalle classi). PARAMETRI indica quali parametri vengono associati ad un metodo (NomePar è il nome del parametro e TipoPar è il tipo del parametro). La cardinalità dell'associazione è Uno-a-Molti (un metodo molti parametri, i parametri non sono condivisi dai metodi).

Con riferimento ad OQL, definire gli EXTENT per i tipi oggetto (non è richiesta la definizione completa). Si scriva una interrogazione che fornisca, per ogni classe indicata dal nome, l'insieme dei nomi dei suoi metodi che operano un override (stesso nome, diversa segnatura) sui metodi della classe padre e l'insieme dei nomi dei metodi che operano un overload (stesso nome, stessa segnatura) sui metodi della classe padre. Si scriva anche il tipo del risultato della interrogazione.

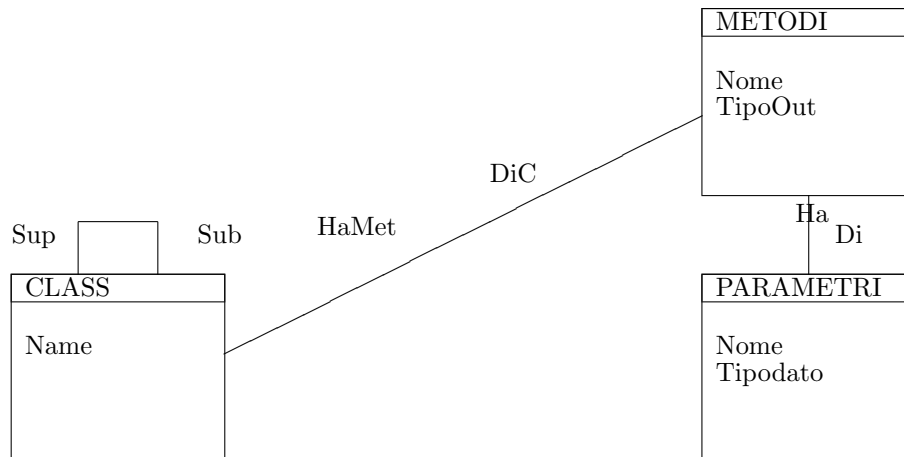


Figure 2: Figura 2

Esercizio 16 Si consideri la bozza di Class Diagram in figura 3 che descrive una base di dati per la gestione di linee ferroviarie:

la classe *TRATTA* contiene la descrizione delle tratte che collegano una stazione di partenza *StPart*, una stazione di arrivo *StArr* e la distanza tra le due stazioni *Km*. *TRENO* contiene la descrizione dei treni che percorrono un insieme di tratte da una stazione di partenza ad una di arrivo. La classe *TRENO* viene specializzata Considerando treni *EuroStar* e *Notturni*. Dei primi si tiene traccia del numero di posti in carrozza ristorante *PRist* e della presenza di servizio bar (attributo booleano). Dei secondi si tiene traccia del numero di posti letto e di cuccette. La sequenza delle tratte percorse da un treno viene descritta dall'associazione: in particolare l'attributo booleano *Stop* indica se il treno ferma nella stazione di arrivo (se vale *False* il treno passa senza fermarsi nella stazione di arrivo della tratta); viene riportata inoltre l'ora di partenza e di arrivo relativa alla percorrenza della tratta. Con riferimento a *OQL*

- si definiscano i tipi di dati (3 punti);
- si definisca un tipo di dato stazione che fornisce il nome della stazione e la lista ordinata dei treni che fanno fermata in quella stazione. Ciascun elemento della lista ordinata indica il treno, il tipo di treno (*Eurostar* o *notturno*), la lunghezza complessiva del tragitto del treno, l'ora di partenza dalla stazione. L'ordinamento per ora di partenza. Si scriva una interrogazione che restituisca le istanze del tipo fermata sopra definito. (6 punti);

Esercizio 17 Si consideri schema di figura 4 che descrive la usuale strutturazione logica di folder e file in un file system.

FOLDER descrive le cartelle. Una cartella può essere contenuta in un'altra cartella (associazione ricorsiva uno a molti). L'attributo *Visibile* può avere valore *'ALL'* (il suo contenuto è visibile a tutti) oppure *'PRIVATE'* (il suo contenuto è visibile ai soli utenti abilitati). Gli utenti abilitati ad accedere ai vari

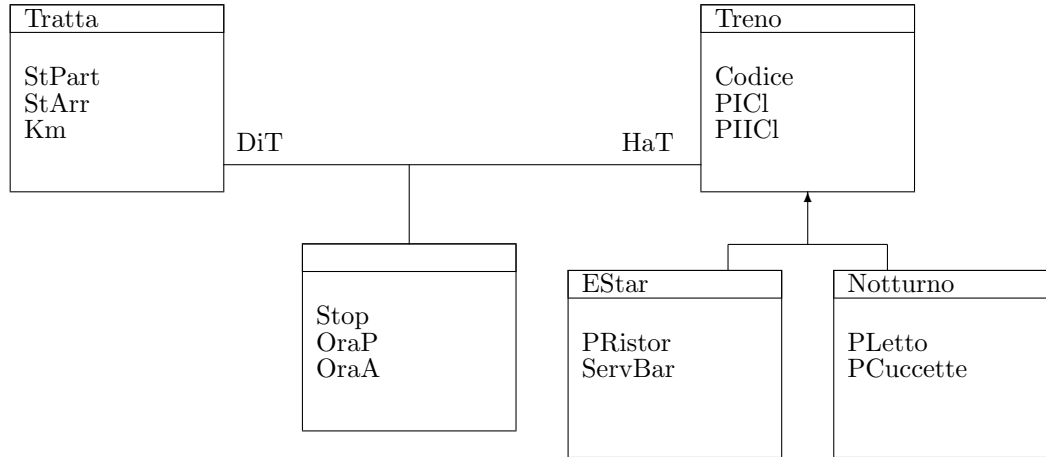


Figure 3:

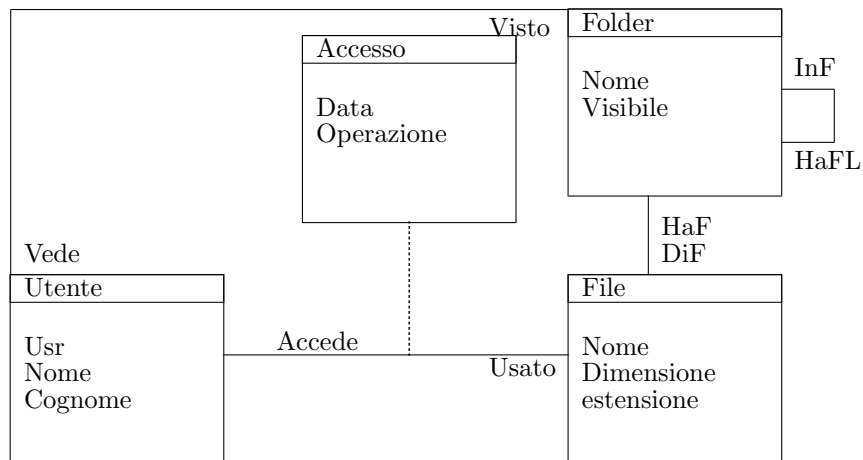


Figure 4: Figura 4

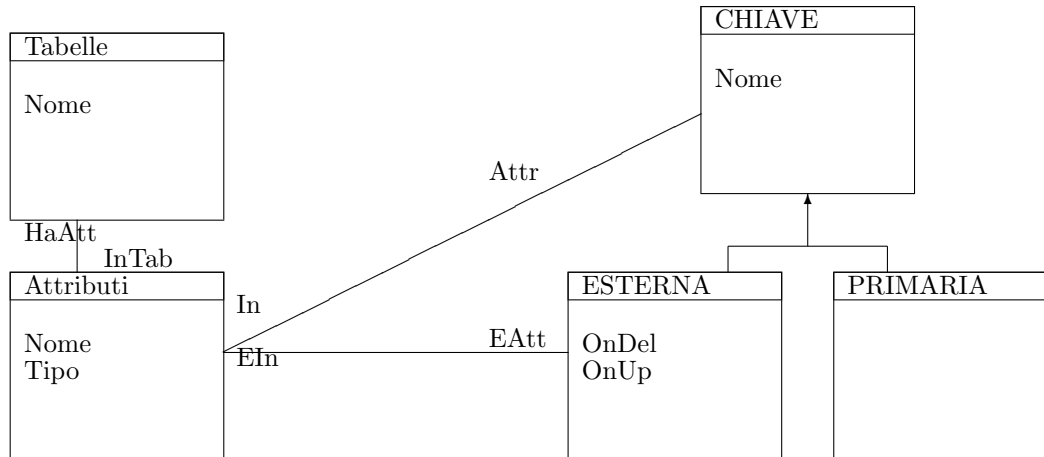


Figure 5: Figura 4

folder sono rappresentati dall'associazione molti a molti. Un utente abilitato ad accedere ad un folder è abilitato ad accedere a tutti i file del folder e a tutti i folder visibili contenuti nel folder. L'associazione tra file e folder è uno a molti. L'accesso degli utenti ai file è descritto dall'associazione molti a molti e l'attributo dell'associazione *Operazione* descrive l'operazione eseguita dall'utente su file nell'accesso avvenuto nella data *Data*.

Con riferimento a OQL dopo aver indicato gli extent (senza entrare necessariamente nel dettaglio dei tipi) si definisca una interrogazione parametrica $F(KK)$ che ha come parametro di ingresso un insieme di folder. L'interrogazione per ogni folder in KK ed ogni data in cui sono state fatte operazioni nei file del folder, associa l'insieme di descrittori utente - nome file (l'utente ha operato nel file). Si usi l'interrogazione $F(KK)$ per trovare la data il cui sono state fatte il maggior numero di operazioni nei file di una delle cartelle visibili della radice della gerarchia delle cartelle (il parametro KK è dunque l'insieme delle cartelle visibili della radice della gerarchia).

Esercizio 18 Si consideri la bozza di Class Diagram in figura 5 che descrive i metadati di un database relazionale. La classe *TABELLA* è associata a uno o più attributi (associazione uno a molti). Una chiave è associata a più attributi (associazione molti a molti). Una chiave è specializzata in chiave primaria o esterna. Se una chiave è specializzata come chiave esterna, l'associazione della specializzazione indica gli attributi della chiave esterna mentre l'associazione della specializzazione indica gli attributi referenziati della tabella esterna.

Senza definire i tipi di dati (ma indicando i nomi degli *EXTENT*) si scriva una interrogazione OQL che associ ad ogni tabella un insieme di descrittori per gli attributi della tabella comprendenti: nome dell'attributo, insieme di coppie *nometabella esterna- attributo tabella esterna*, dove *tabella esterna* e *attributo tabella esterna* sono legati, all'attributo considerato, da un vincolo di chiave esterna (la tabella esterna riferenzia quella considerata). L'interrogazione deve recuperare l'informazione richiesta per la tabella referenziata dal maggior nu-

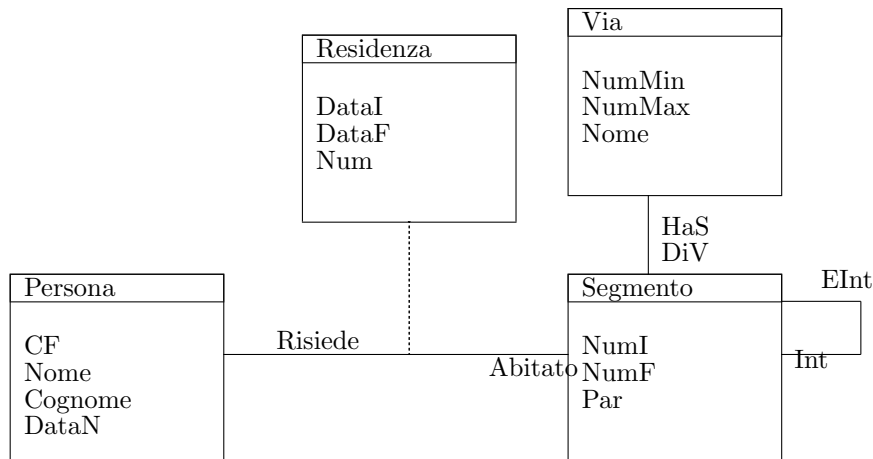


Figure 6: Figura 6

mero di altre tabelle (mediante chiavi esterne).

4 Data Base Relazionale a Oggetti

Esercizio 20 Si consideri la bozza di Class Diagram in figura 1 che descrive la composizione di grafi. Ogni grafo ha un insieme di archi e di nodi. Gli archi e i nodi sono condivisi dai grafi. Un albero è la specializzazione di un grafo. Le etichette di grafi e nodi sono numeri interi.

Con riferimento al modello relazionale ad oggetti di Oracle

- si implementi in modo ragionato il Class Diagram di figura 1 definendo tipi e tabelle;
- si aggiunga un metodo PL-SQL al tipo GRAFO che associa un arco all'insieme degli archi del grafo (il REF dell'arco da associare viene passato come parametro). Il metodo controlla che i nodi sorgente e destinazione dell'arco facciano parte dell'insieme dei nodi del grafo (se non fanno parte, essi vengono inseriti a loro volta).
- Si stabilisca un metodo che ordini i tipi oggetto albero in base al cammino di peso massimo (somma dei pesi degli archi). Per semplicità, si suggerisce che per calcolare i cammini di peso massimo si proceda a scandire il cammino dalle foglie alla radice.

Esercizio 21 Con riferimento al modello object-relational di Oracle, si definiscano tipi e tabelle per il class diagram di figura 2. Nella definizione si chiede che i metodi siano memorizzati come sottotabella delle classi e che i parametri siano memorizzati come sottotabella dei metodi.

Si aggiunga alla definizione di CLASS il metodo `AggiungiPar(NomeMet, NomePar, TipoPar)`. Se un metodo di nome `Nomemet` è già associato alla classe corrente, `AggiungiPar`

Esercizio 19 Si consideri la bozza di Class Diagram in figura 6 che descrive lo storico delle residenze delle persone nei segmenti di via (che delimitano isolati non considerati per semplicità). Un segmento di via è una porzione di lato di una via caratterizzato da un numero civico inferiore, uno superiore e dalla parità dei numeri. Una associazione molti a molti tra i segmenti indica la relazione di intersezione (incrocio) tra i segmenti di via. La residenza è caratterizzata da una data di inizio e una di fine (attributo parziale, nullo per le residenze ancora in corso). Si osservi che la cardinalità dell'associazione tra Persona e Segmento è molti a molti.

Con riferimento a OQL:

- Avendo definito gli extent dei tipi di dati necessari (non è necessaria la definizione esplicita dei tipi), si scriva una interrogazione OQL che per ogni via fornisca l'insieme dei nomi delle via che la intersecano e il nome della via con maggior numero di residenze correnti tra quelle che la intersecano.
- si scriva il tipo disponibile nella clausola select della seguente interrogazione e si completi poi la clausola select in modo che venga riportato per ogni anno il numero delle residenze terminate e l'insieme contenente il nome e cognome delle residenze terminate.

```
SELECT
FROM V AS VIA, K AS V.has, R AS K.Abitato
GROUP BY
N : R.DataI.Year
Z : isundefined(R.DataF)
```

Figure 7: Figura 7

Figure 8: Figura 8

Figure 9: Figura 4

aggiunge il parametro *NomePar* di tipo *TipoPar* al metodo, altrimenti, prima crea il nuovo metodo con *TipoOut* settato a *Void* e poi vi aggiunge il parametro. Si aggiunga alla definizione di *CLASS* un metodo che restituisce una stringa che contiene la lista dei nomi dei metodi visibili in quella classe (i nomi visibili sono quelli associati alla classe stessa e a tutte le sue superclassi).

Esercizio 22 Si consideri la bozza di *Class Diagram* in figura 3 che descrive una base di dati per la gestione di linee ferroviarie:

la classe *TRATTA* contiene la descrizione delle tratte che collegano una stazione di partenza *StPart*, una stazione di arrivo *StArr* e la distanza tra le due stazioni *Km*. *TRENO* contiene la descrizione dei treni che percorrono un insieme di tratte da una stazione di partenza ad una di arrivo. La classe *TRENO* viene specializzata Considerando treni *EuroStar* e *Notturni*. Dei primi si tiene traccia del numero di posti in carrozza ristorante *PRist* e della presenza di servizio bar (attributo booleano). Dei secondi si tiene traccia del numero di posti letto e di cuccette. La sequenza delle tratte percorse da un treno viene descritta dall'associazione: in particolare l'attributo booleano *Stop* indica se il treno ferma nella stazione di arrivo (se vale *False* il treno passa senza fermarsi nella stazione di arrivo della tratta); viene riportata inoltre l'ora di partenza e di arrivo relativa alla percorrenza della tratta. Con riferimento al modello relazionale ad oggetti di *Oracle*, si integri lo schema precedente con un metodo *Percorre* da associare alla classe *treno* che prende in ingresso un *REF* ad una tratta, un valore booleano per l'attributo *Stop*, un'ora di partenza ed un'ora di arrivo. Il metodo inserisce una istanza di associazione tra la tratta passata come parametro e l'istanza di treno corrente.

- Si implementi in modo ragionato il *Class Diagram* sopra descritto ;
- Si implementi in *PL-SQL* il metodo *Percorre*;
- Si scriva una interrogazione in *SQL* che restituisca codice del treno, tipologia (*Eurostar* o *Notturmo*) per treni che fermano in tutte le stazioni attraversate.

Esercizio 23 Si consideri schema di figura 4 che descrive la usuale strutturazione logica di folder e file in un file system.

FOLDER descrive le cartelle. Una cartella può essere contenuta in un'altra cartella (associazione ricorsiva uno a molti). L'attributo *Visibile* può avere valore '*ALL*' (il suo contenuto è visibile a tutti) oppure '*PRIVATE*' (il suo contenuto è visibile ai soli utenti abilitati). Gli utenti abilitati ad accedere ai vari folder sono rappresentati dall'associazione molti a molti. Un utente abilitato ad accedere ad un folder è abilitato ad accedere a tutti i file del folder e a tutti i folder visibili contenuti nel folder. L'associazione tra file e folder è uno a molti. L'accesso degli utenti ai file è descritto dall'associazione molti a molti e l'attributo dell'associazione *Operazione* descrive l'operazione eseguita dall'utente su file nell'accesso avvenuto nella data *Data*.

Con riferimento al relazionale ad oggetti di *Oracle*

- si definiscano sommariamente i tipi (i nomi la eventuale sottostruttura di ref e collezioni) e le tabelle per il diagramma;
- si scriva un metodo da associare alla classe *Utente* che riceve in ingresso il riferimento ad un *File* e un tipo di operazione; l'effetto è di creare una istanza di accesso dell'utente al file.
- si definisca un metodo da associare alla classe *Folder* che restituisce lo spazio di memoria occupato dai file contenuti nel folder (a tutti i livelli). Si usi per determinare la visita della gerarchia una tabella ausiliaria *TEMP* per memorizzare i risultati intermedi.

Esercizio 24 Si consideri la bozza di *Class Diagram* in figura 5 che descrive i metadati di un database relazionale. La classe *TABELLA* è associata a uno o più attributi (associazione uno a molti). Una chiave è associata a più attributi (associazione molti a molti). Una chiave è specializzata in chiave primaria o esterna. Se una chiave è specializzata come chiave esterna, l'associazione della superclasse indica gli attributi della chiave esterna mentre l'associazione della specializzazione indica gli attributi referenziati della tabella esterna. Sfruttando le caratteristiche object relational di Oracle, si scelga una implementazione del *Class Diagram* sopra descritto.

Si aggiunga alla classe *TABELLE* un metodo opportuno per l'ordinamento delle istanze di tabelle. Il criterio per l'ordinamento prevede che una tabella *T1* precede la tabella *T2* se esiste una chiave esterna in *T2* che riferenzia *T1*.

Si aggiunga alla classe *CHIAVE* una funzione booleana che controlli che tutti gli attributi referenziati da una chiave esterna costituiscano una chiave primaria nella tabella referenziata.

Esercizio 25 Si consideri la bozza di *Class Diagram* in figura 6 che descrive lo storico delle residenze delle persone nei segmenti di via (che delimitano isolati non considerati per semplicità). Un segmento di via è una porzione di lato di una via caratterizzato da un numero civico inferiore, uno superiore e dalla parità dei numeri. Una associazione molti a molti tra i segmenti indica la relazione di intersezione (incrocio) tra i segmenti di via. La residenza è caratterizzata da una data di inizio e una di fine (attributo parziale, nullo per le residenze ancora in corso). Si osservi che la cardinalità dell'associazione tra *Persona* e *Segmento* è molti a molti.

Con riferimento al modello relazionale ad oggetti di Oracle si indichino qualitativamente (elencando i nomi anche senza definizione esplicita) i tipi di dati necessari alla codifica del problema e si fornisca la definizione delle tabelle (sfruttando i nomi dei tipi informalmente definiti).

- si aggiunga un metodo *PL-SQL* al tipo *persona* per il cambio di residenza. Il metodo riceve come parametri il nome della via e il numero civico. Il metodo sortisce come effetto la chiusura della vecchia residenza e l'inserimento della nuova.

- *si scriva una interrogazione che recuperi nome e cognome delle persone che hanno fatto un cambio di residenza da una via ad una via ad essa intersecante.*