Corso di Basi di dati
Prova scritta parziale
1 marzo 2001

Tempo a disposizione: un'ora e trenta minuti.
Libri e appunti chiusi.

Soluzioni

Sono riportate le soluzioni per il compito A. Quelle del compito B sono molto simili.

Soluzione alla domanda 1

1. L'anomalia è di tipo "lost update"; esempio di schedule (il volo deve essere lo stesso, il numero di posti sarà in generale diverso (quindi si usa lo stesso x e due diversi y_1 e y_2):

	T_i	T_j	*valore di x*
1.	*read(x)*		x_0
2.		*read(x)*	x_0
3.	*x:= x - y_1*		x_0
4.	*write(x)*		$x_0 - y_1$
5.		*x:= x - y_2*	$x_0 - y_1$
6.		*write(x)*	$x_0 - y_2$

2. Il controllo di concorrenza basato su timestamp, alla riga 4, rileva che la transazione T_i (che assumiamo meno giovane di T_j) vuole scrivere un dato già letto da una transazione più giovane. Pertanto, T_i viene uccisa.
3. Attraverso l'emissione di richieste di lock e le regole che governano i lock stessi, le azioni vengono riordinate:

	T_i	T_j	*lock manager*	stato di x	*valore di x*
1.	*writelock(x)*				x_0
2.			*writelockOK(T_i,x)*	*w-locked(T_i)*	x_0
3.	*read(x)*			*w-locked(T_i)*	x_0
4.		*writelock(x)*		*w-locked(T_i)*	x_0
5.			*wait(T_j,x)*	*w-locked(T_i)*	x_0
6.	*x:= x - y_1*			*w-locked(T_i)*	x_0
7.	*write(x)*			*w-locked(T_i)*	$x_0 - y_1$
8.	*commit*			*w-locked(T_i)*	$x_0 - y_1$
9.	*unlock(x)*			*libera*	$x_0 - y_1$
10.			*writelockOK(T_j,x)*	*w-locked(T_j)*	$x_0 - y_1$
11.		*read(x)*		*w-locked(T_j)*	$x_0 - y_1$
12.		*x:= x - y_2*		*w-locked(T_j)*	$x_0 - y_1$
13.		*write(x)*		*w-locked(T_j)*	$x_0 - y_1 - y_2$
14.		*commit*		*w-locked(T_j)*	$x_0 - y_1 - y_2$
15.		*unlock(x)*		*libera*	$x_0 - y_1 - y_2$

Nell'esempio, sono stati utilizzati solo lock in scrittura (caso (b)). Con lock in lettura, da promuovere, si sarebbero potuti avere deadlock. In generale, la scelta (a) è più efficiente (ad esempio, se ci sono altre operazioni di sola lettura, esse possono procedere, almeno nella prima fase), ma presenta il rischio dello stallo, che invece non si corre nel caso (b).

Soluzione alla domanda 2

È sufficiente prevedere solo redo: poiché prima del commit non ci sono scritture, non ci sono operazioni da annullare per transazioni non andate a buon fine.

Questa procedura non viene molto utilizzata in pratica perché, pur più efficiente nel recovery, è complessivamente meno efficiente di una in cui il gestore del buffer può decidere liberamente quando scrivere in memoria secondaria. Si preferisce cioè una gestione ordinaria più efficiente rispetto ad una gestione più semplice dei guasti, poiché si assume che i guasti siano abbastanza rari.

Soluzione alla domanda 3
Vedi le figure alle pagine seguenti.

- il file ha fattore di blocco (cioè numero di record per blocco) pari a 4 (parte intera inferiore di 500/110) e quindi occupa 2500 blocchi
- l'indice primario su Cognome può essere sparso e in tal caso ha
  - record di 24 caratteri ciascuno e quindi un fattore di blocco pari a 20 (pari a 500/24)
  - al livello più basso, 2500 record (uno per ciascun blocco del file da indicizzare in quanto l'indice è sparso) e quindi 125 blocchi (2500/20)
  - al livello superiore, 125 record e quindi 7 blocchi (125/20)
  - 7 record in un solo blocco al livello ulteriormente superiore

  In totale occupa quindi circa 133 blocchi.
- l'indice secondario su Matricola deve essere denso e ha
  - record di 8 caratteri e quindi fattore di blocco massimo pari a 62, ma poiché si tratta di un B-tree il riempimento è parziale; assumendo circa il 70%, abbiamo mediamente circa 40 record per blocco
  - 10000 record al livello più basso (perché denso), in circa 250 blocchi (10000/40)
  - 250 record al livello superiore, in 7 blocchi
  - un solo blocco al livello ulteriormente superiore

  In totale quindi potremmo avere circa 260 blocchi.

Soluzione alla domanda 4 Vedi la terza figura. L'indice secondario hash ha record di 8 caratteri come quelli del B-tree (valore della chiave e puntatore); con un riempimento pari all'80%, si possono avere circa 50 record per blocco e quindi l'indice può avere 200 blocchi e quindi la funzione hash può essere il resto della divisione del numero di matricola per 200.

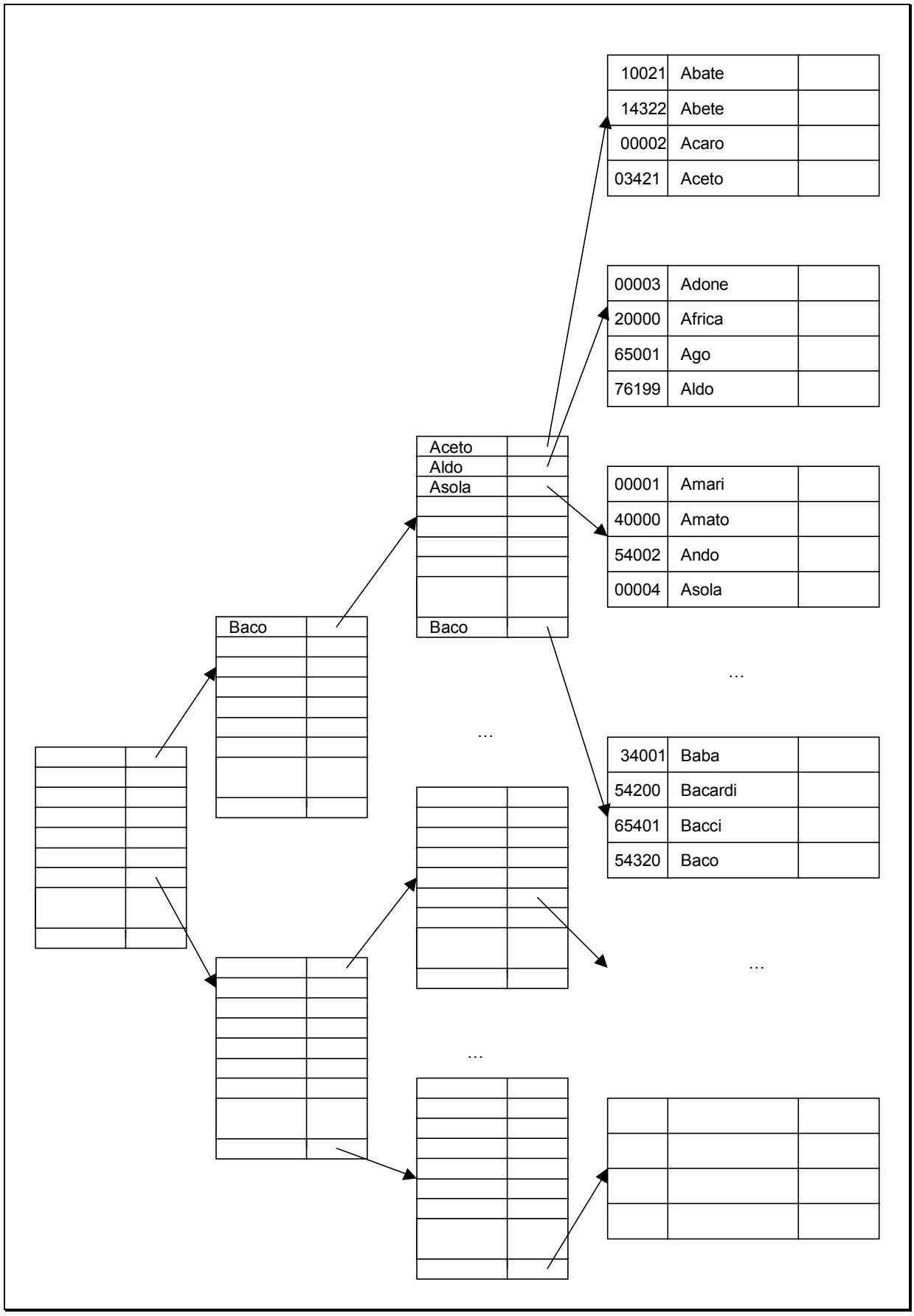
10021 Abate
14322 Abete
00002 Acaro
03421 Aceto
00003 Adone
20000 Africa
65001 Ago
76199 Aldo
Aceto
Aldo
Asola
Baco
00001 Amari
40000 Amato
54002 Ando
00004 Asola
...
Baco
...
34001 Baba
54200 Bacardi
65401 Bacci
54320 Baco
...
...

Figura 1

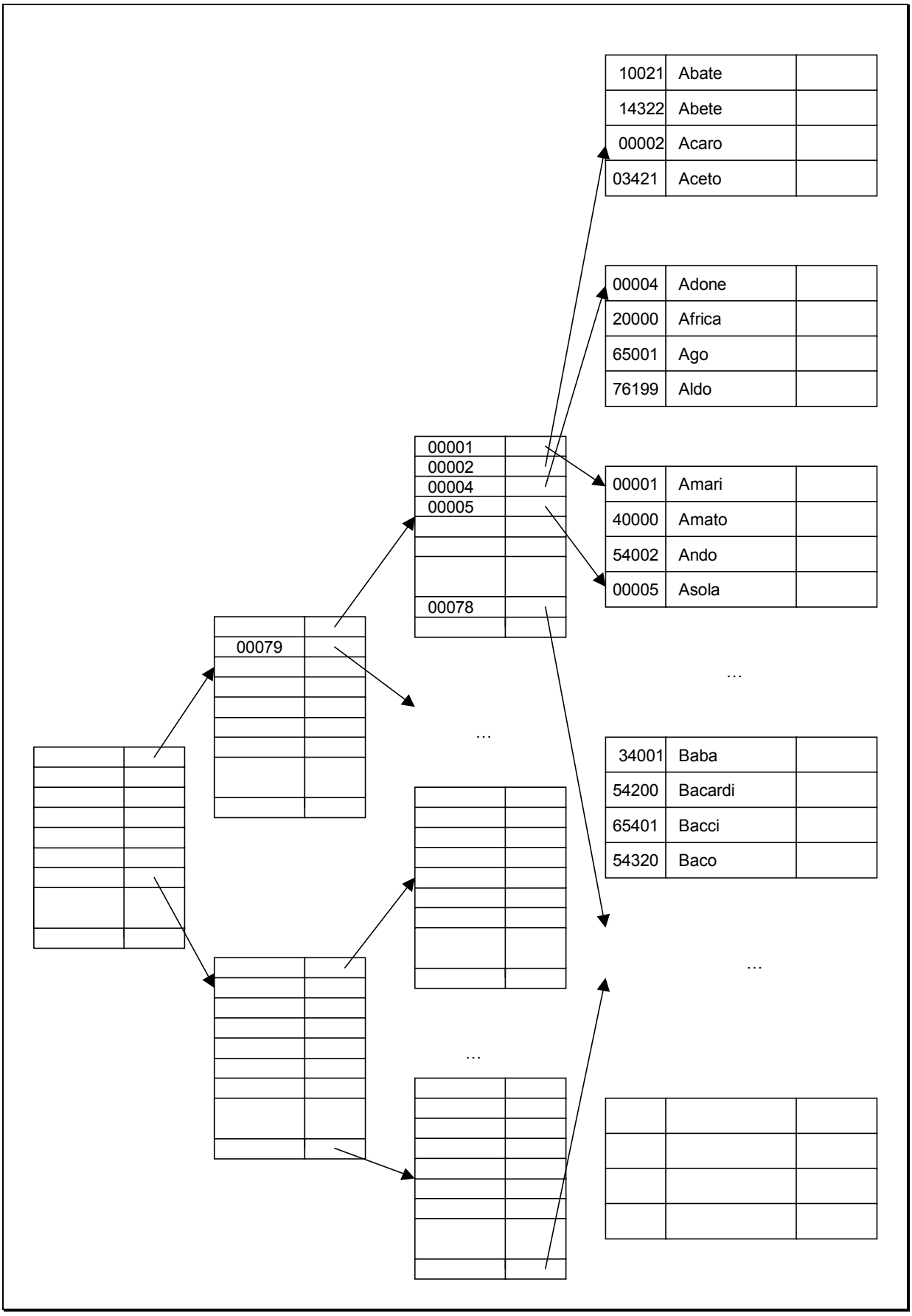
10021 Abate
14322 Abete
00002 Acaro
03421 Aceto
00004 Adone
20000 Africa
65001 Ago
76199 Aldo
00001
00002
00004
00005
00078
00001 Amari
40000 Amato
54002 Ando
00005 Asola
00079
...
...
34001 Baba
54200 Bacardi
65401 Bacci
54320 Baco
...
...

Figura 2

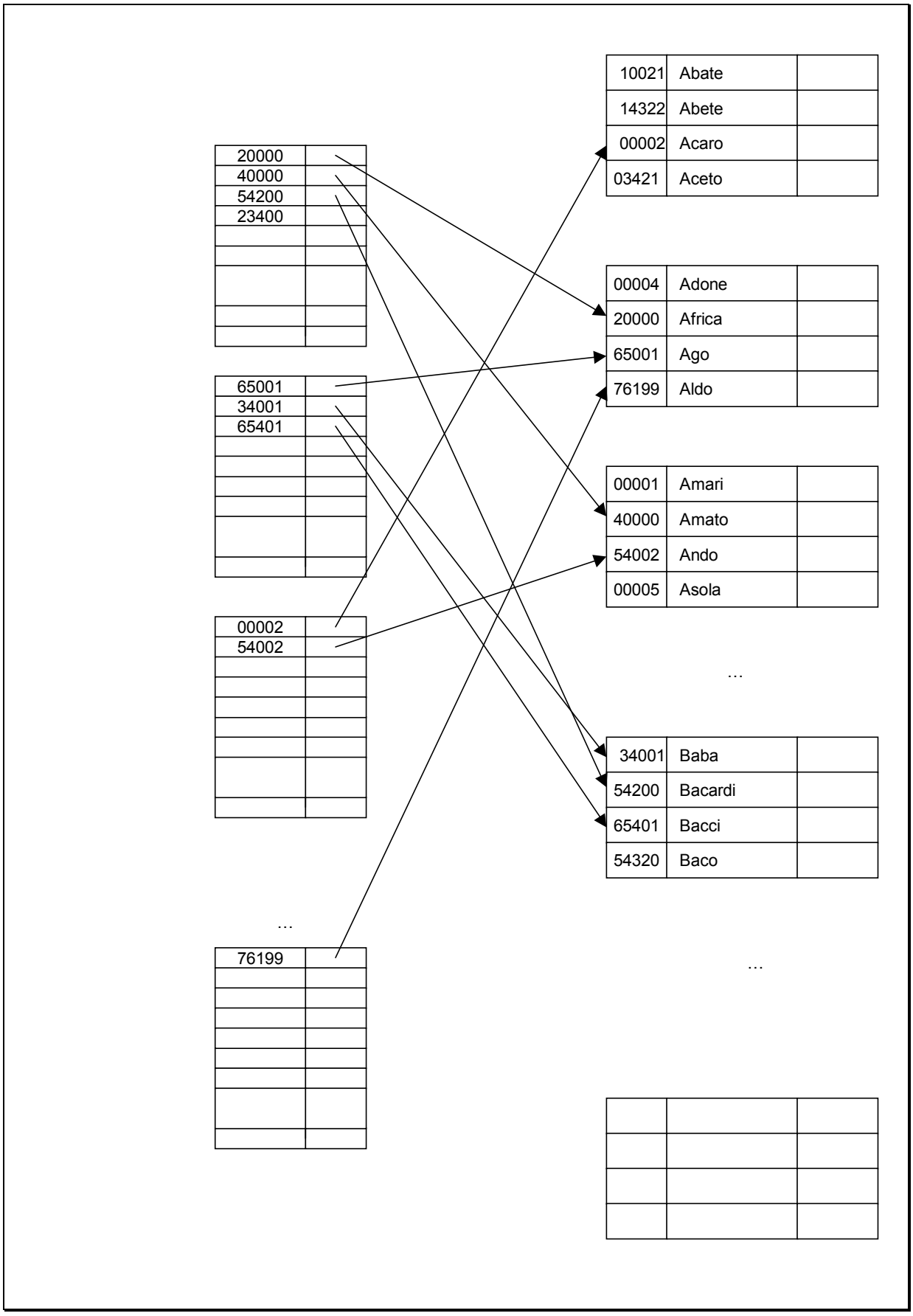
20000
40000
54200
23400
65001
34001
65401
00002
54002
...
76199
10021 Abate
14322 Abete
00002 Acaro
03421 Aceto
00004 Adone
20000 Africa
65001 Ago
76199 Aldo
00001 Amari
40000 Amato
54002 Ando
00005 Asola
...
34001 Baba
54200 Bacardi
65401 Bacci
54320 Baco
...

Figura 3