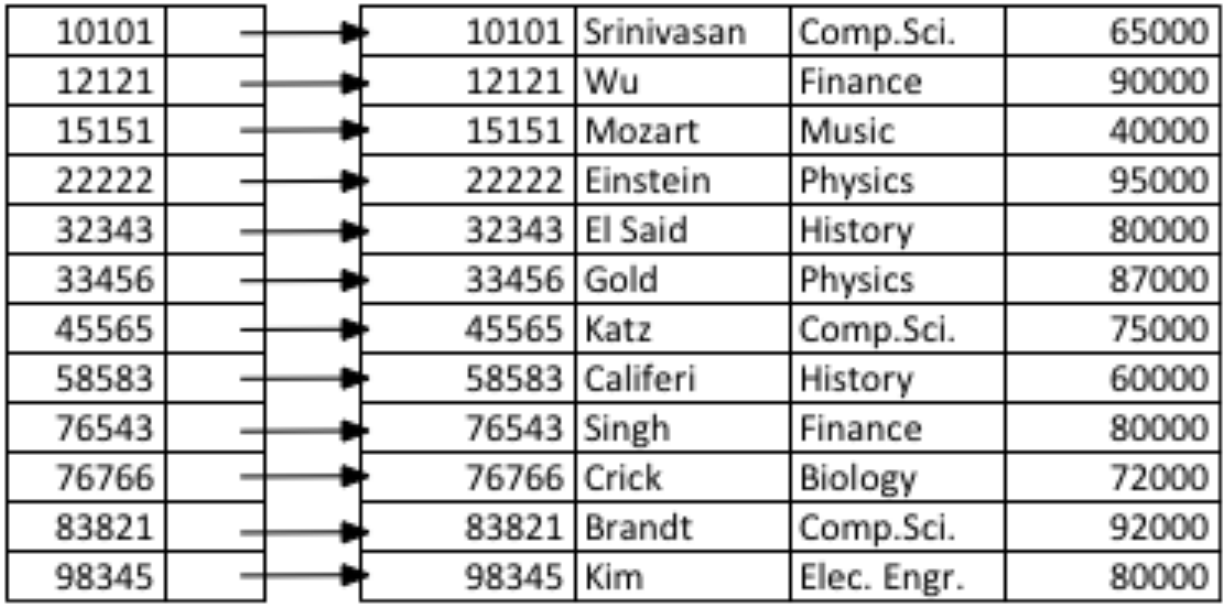
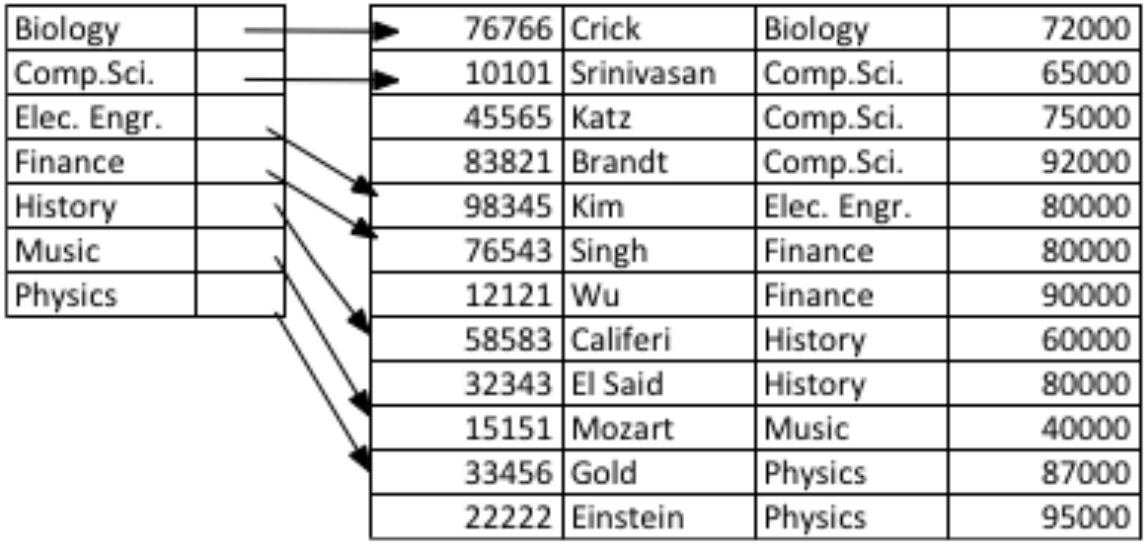
Dense Index



10101	10101	Srinivasan	Comp.Sci.	65000
12121	12121	Wu	Finance	90000
15151	15151	Mozart	Music	40000
22222	22222	Einstein	Physics	95000
32343	32343	El Said	History	80000
33456	33456	Gold	Physics	87000
45565	45565	Katz	Comp.Sci.	75000
58583	58583	Califeri	History	60000
76543	76543	Singh	Finance	80000
76766	76766	Crick	Biology	72000
83821	83821	Brandt	Comp.Sci.	92000
98345	98345	Kim	Elec. Engr.	80000

From Silberschatz, et.al., *Database Systems Concepts*, 6th ed

Dense Index sorted on other than primary key

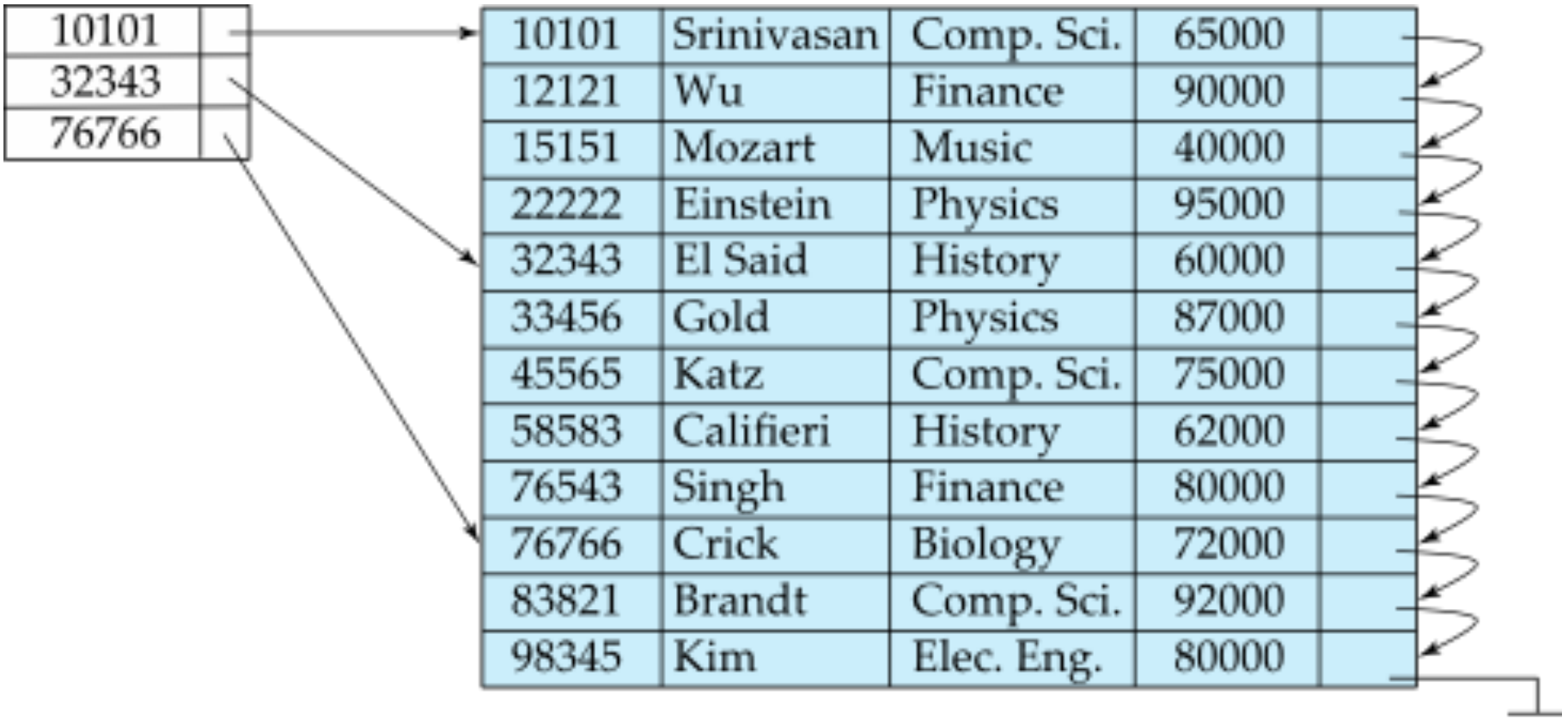


Biology	
Comp.Sci.	
Elec. Engr.	
Finance	
History	
Music	
Physics	

76766	Crick	Biology	72000
10101	Srinivasan	Comp.Sci.	65000
45565	Katz	Comp.Sci.	75000
83821	Brandt	Comp.Sci.	92000
98345	Kim	Elec. Engr.	80000
76543	Singh	Finance	80000
12121	Wu	Finance	90000
58583	Califeri	History	60000
32343	El Said	History	80000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
22222	Einstein	Physics	95000

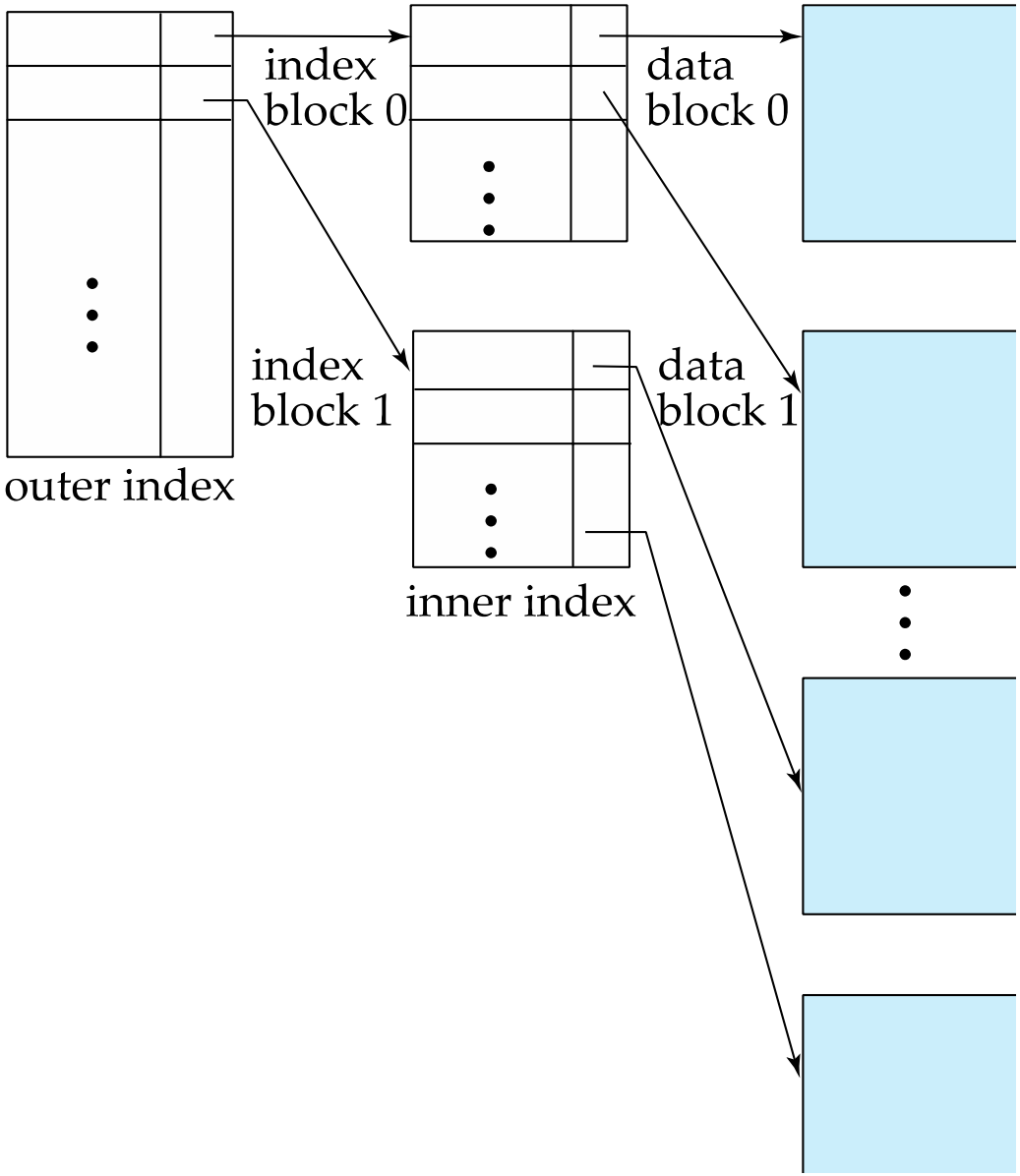
From Silberschatz, et.al., *Database Systems Concepts*, 6th ed.

Sparse index

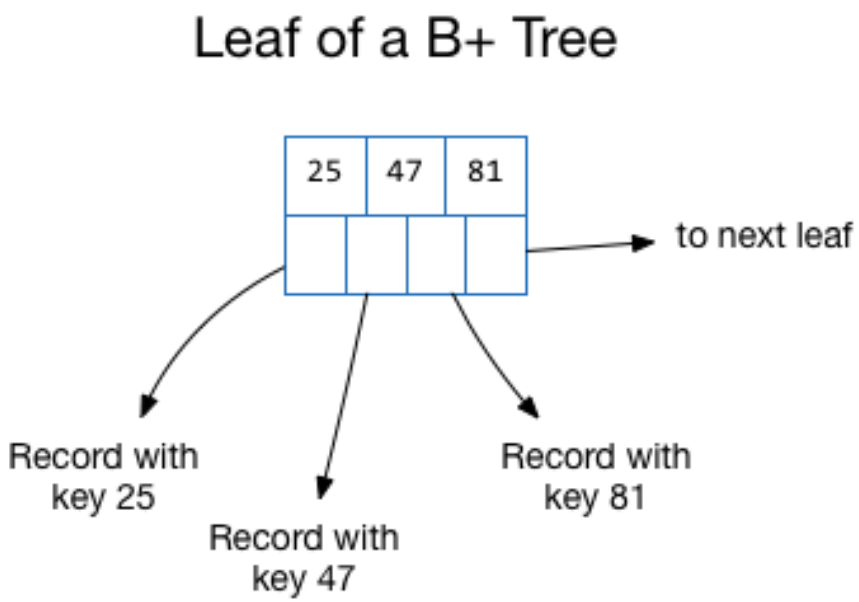


From Silberschatz, et.al., *Database Systems Concepts*, 6th ed.

Multi-Level Index

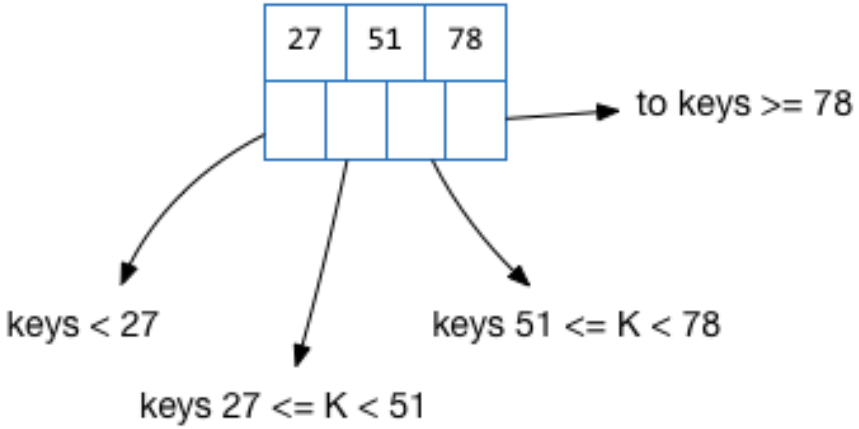


From Silberschatz, et.al., *Database Systems Concepts*, 6th ed.



From Garcia-Molina, *DB Systems: The Complete Book*

Interior node of a B+ Tree



From Garcia-Molina, *DB Systems: The Complete Book*

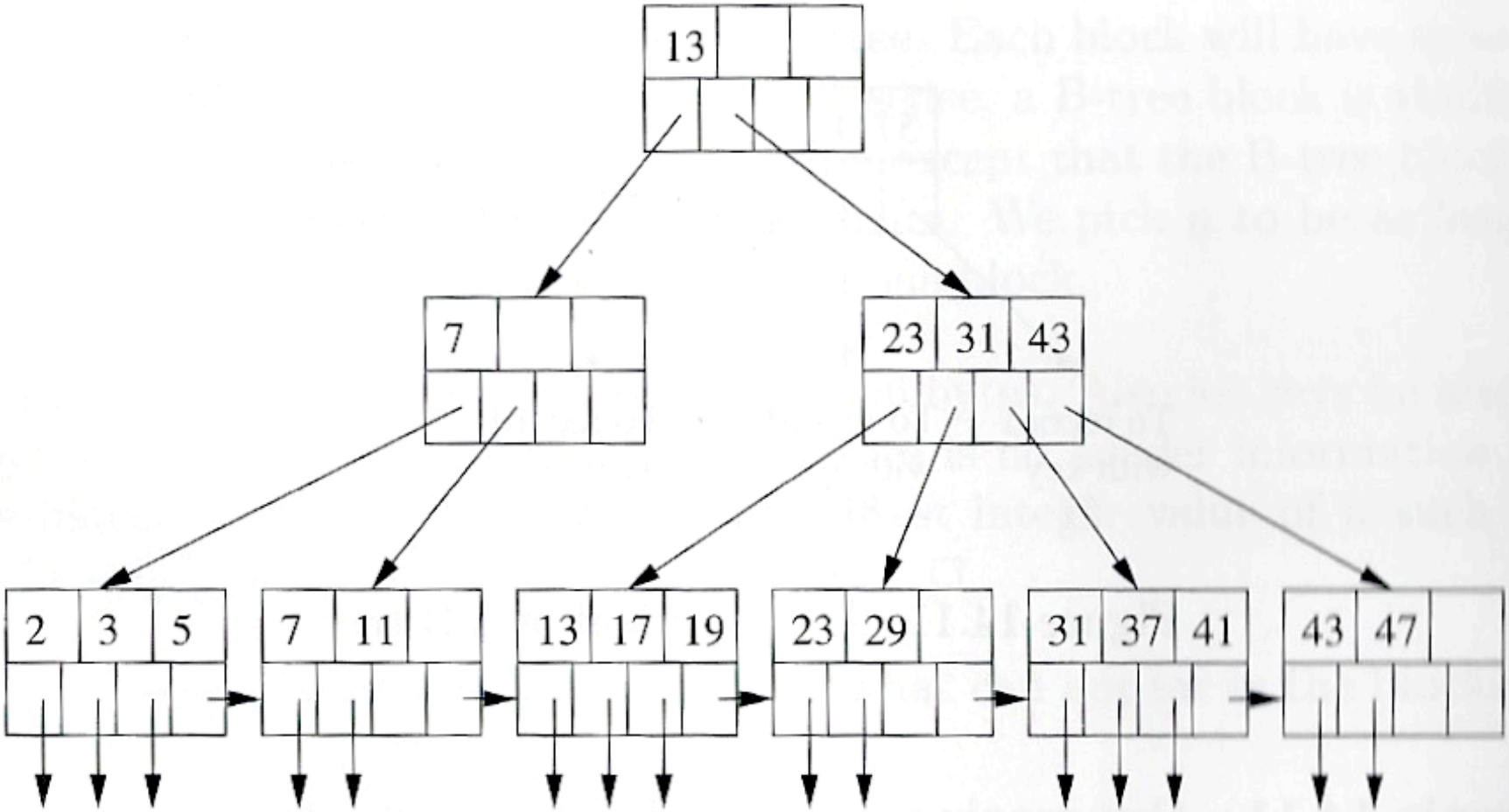


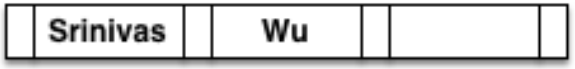
Figure 14.13: A B-tree

From Garcia-Molina, *DB Systems: The Complete Book*

Example of B+-Tree - building from the instructor table

Step 1. Insert Srinivas

Step 2. Insert Wu

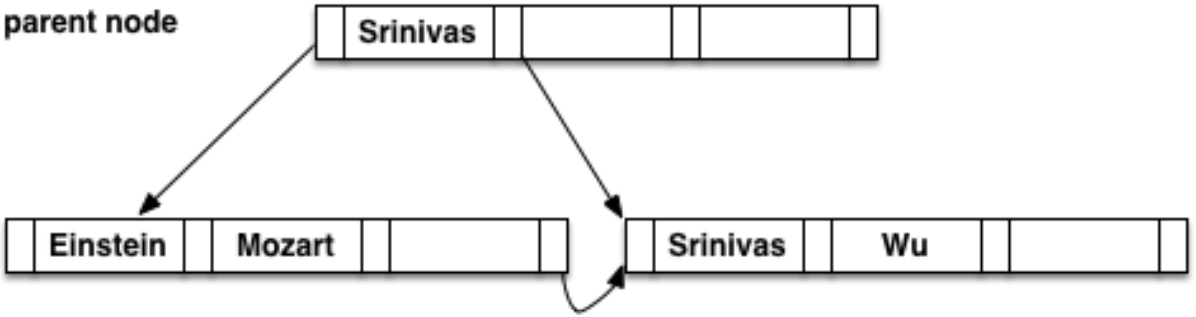


Step 3. Insert Mozart

Step 4. Insert Einstein

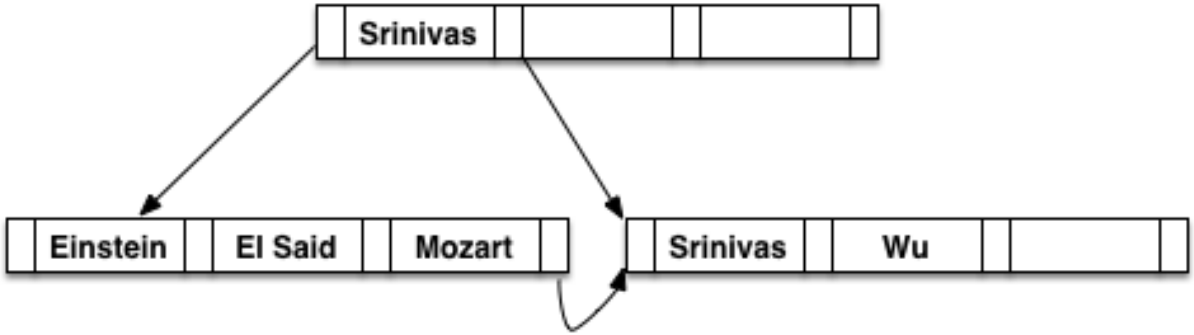
Step 4a. Split the root node

Step 4b. Create a new parent node



From Silberschatz, et.al., *Database Systems Concepts*, 6th ed.

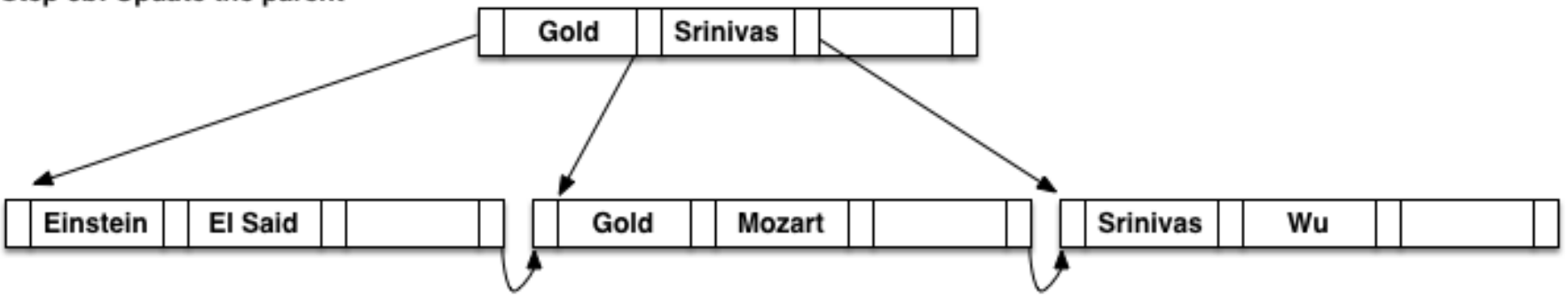
Step 5. Insert El Said



Step 6. Insert Gold

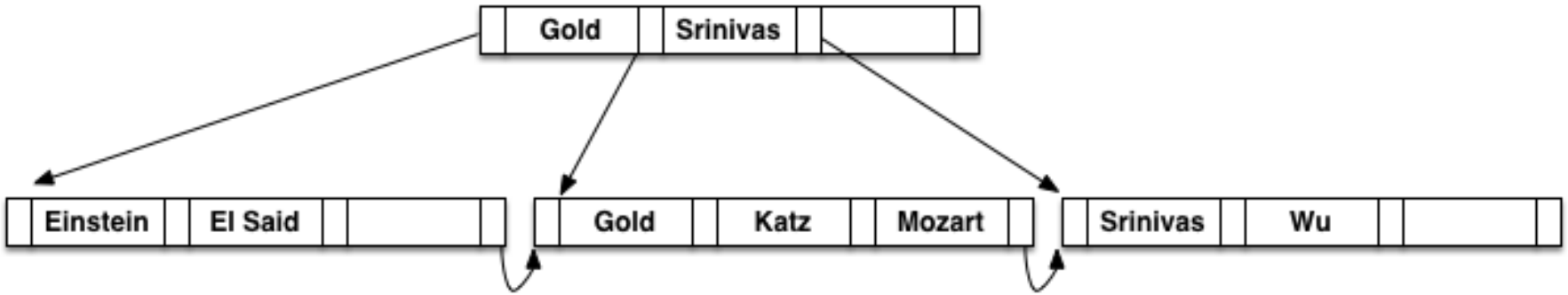
Step 6a: Overflowed the node, so split it

Step 6b: Update the parent

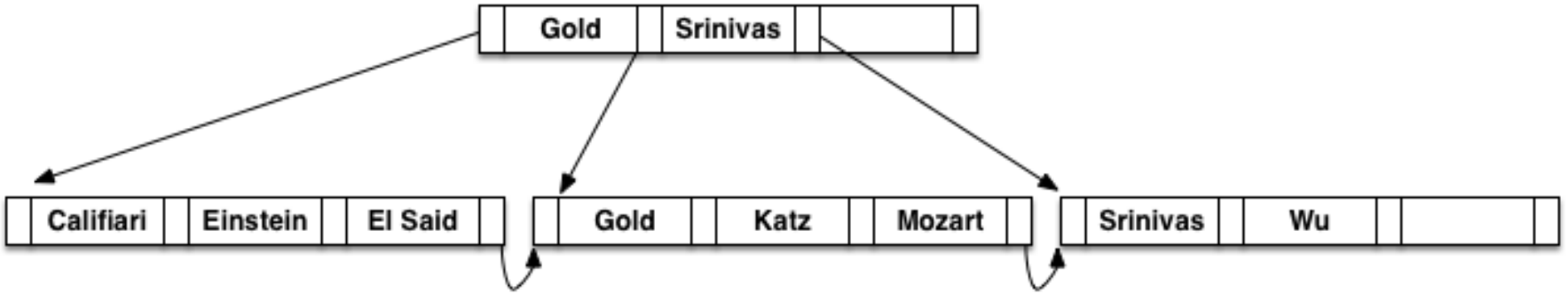


From Silberschatz, et.al., *Database Systems Concepts*, 6th ed.

Step 7. Insert Katz



Step 8. Insert Califieri

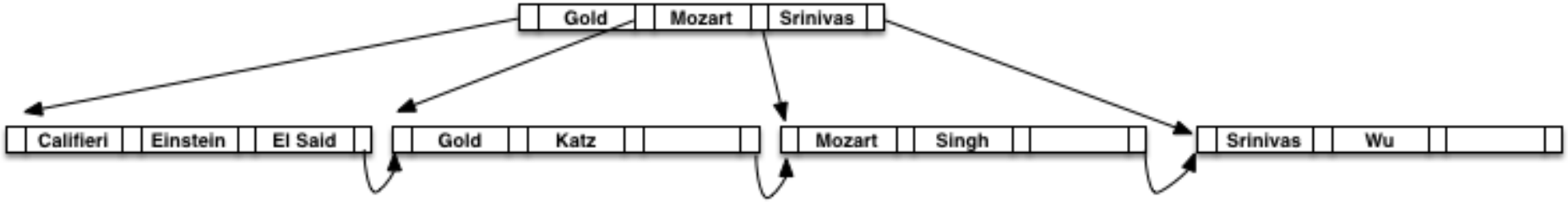


From Silberschatz, et.al., *Database Systems Concepts*, 6th ed.

Step 9: Insert Singh

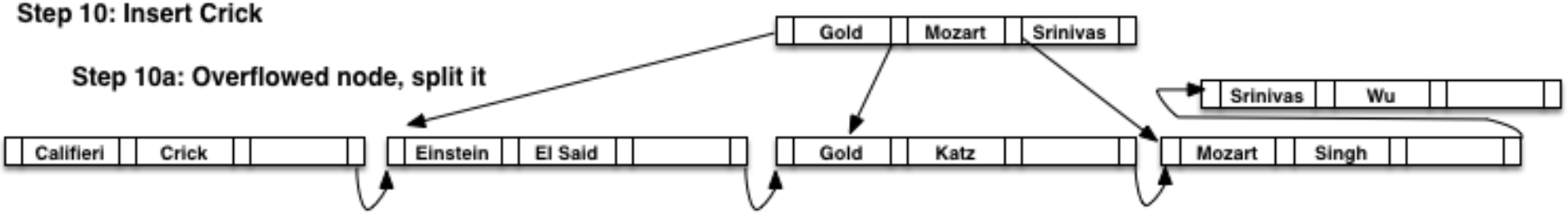
Step 9a: overflowed the node, so split it

Step 9b: update parent

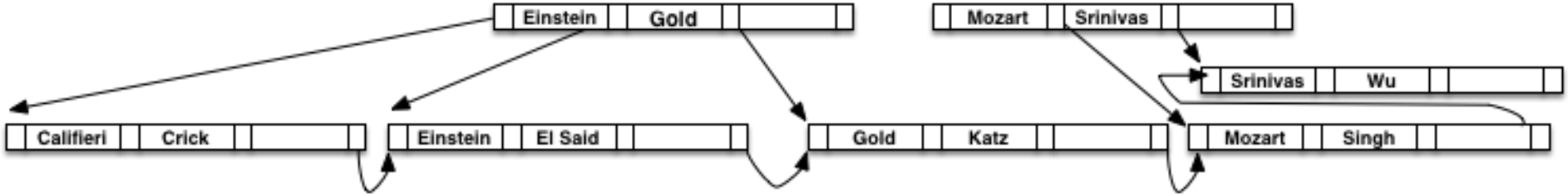


Step 10: Insert Crick

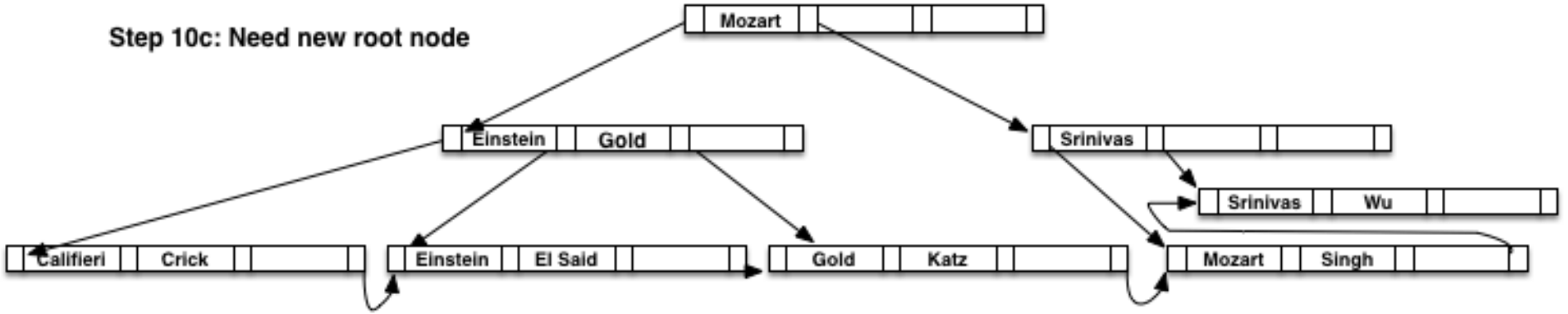
Step 10a: Overflowed node, split it

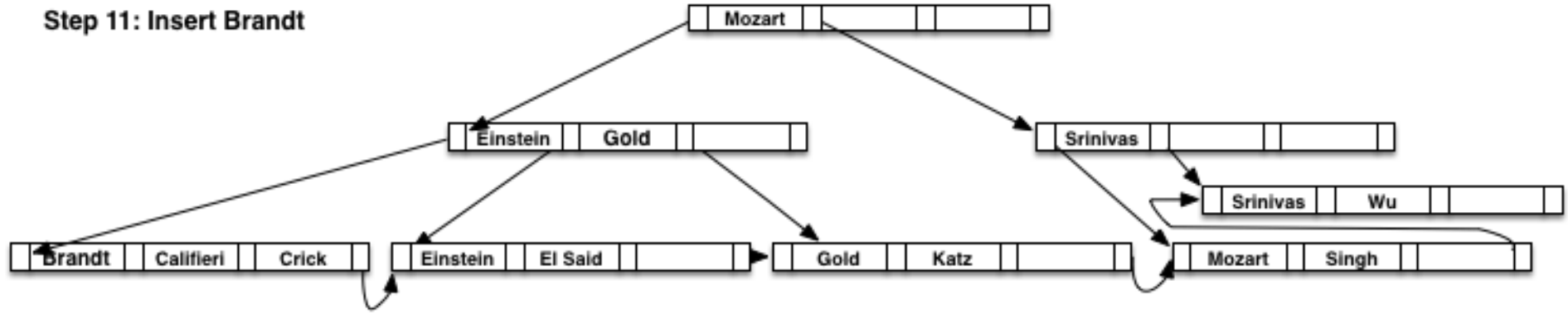


Step 10b: Update parent, but overflowed parent node, split it

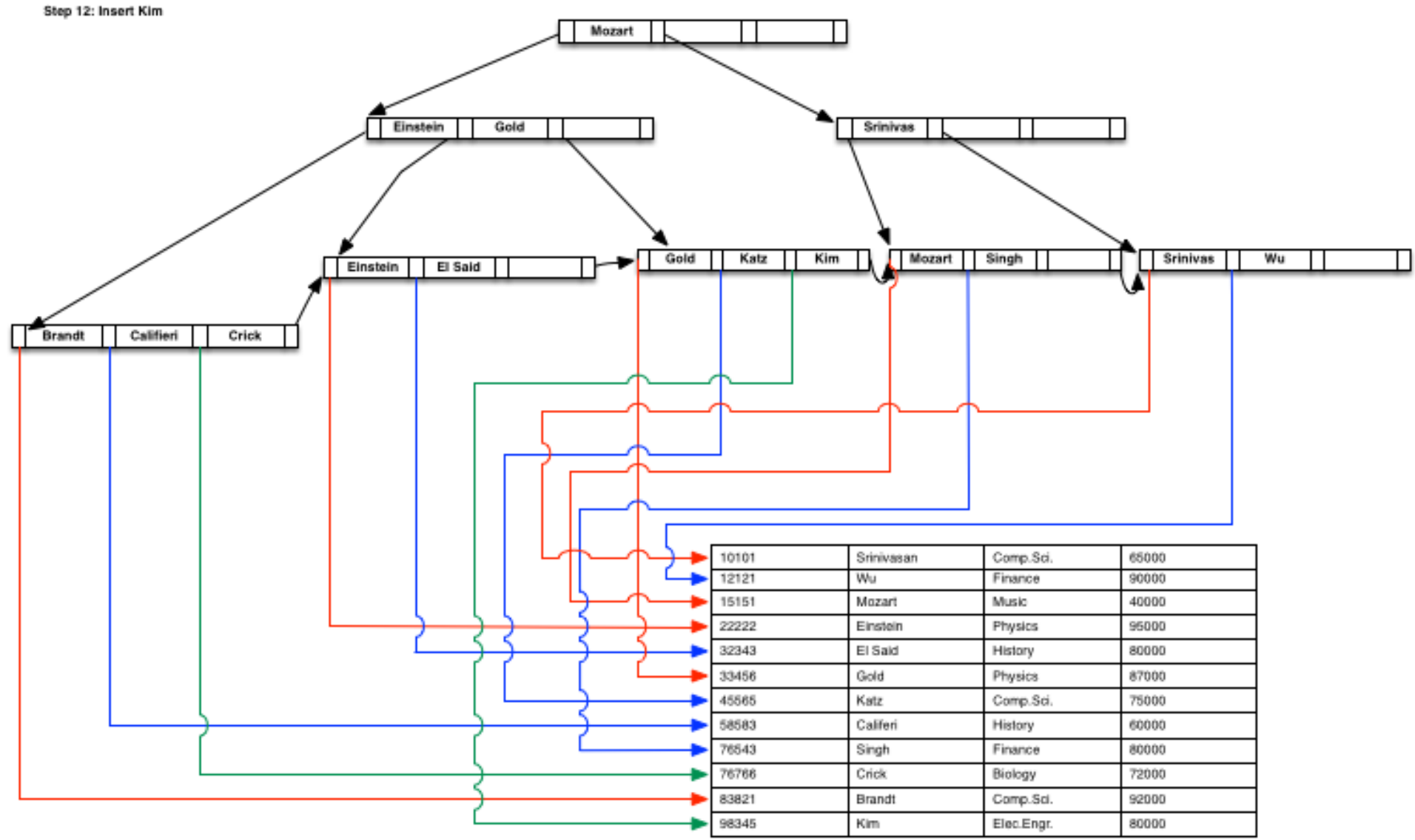


From Silberschatz, et.al., *Database Systems Concepts*, 6th ed.





From Silberschatz, et.al., *Database Systems Concepts*, 6th ed.



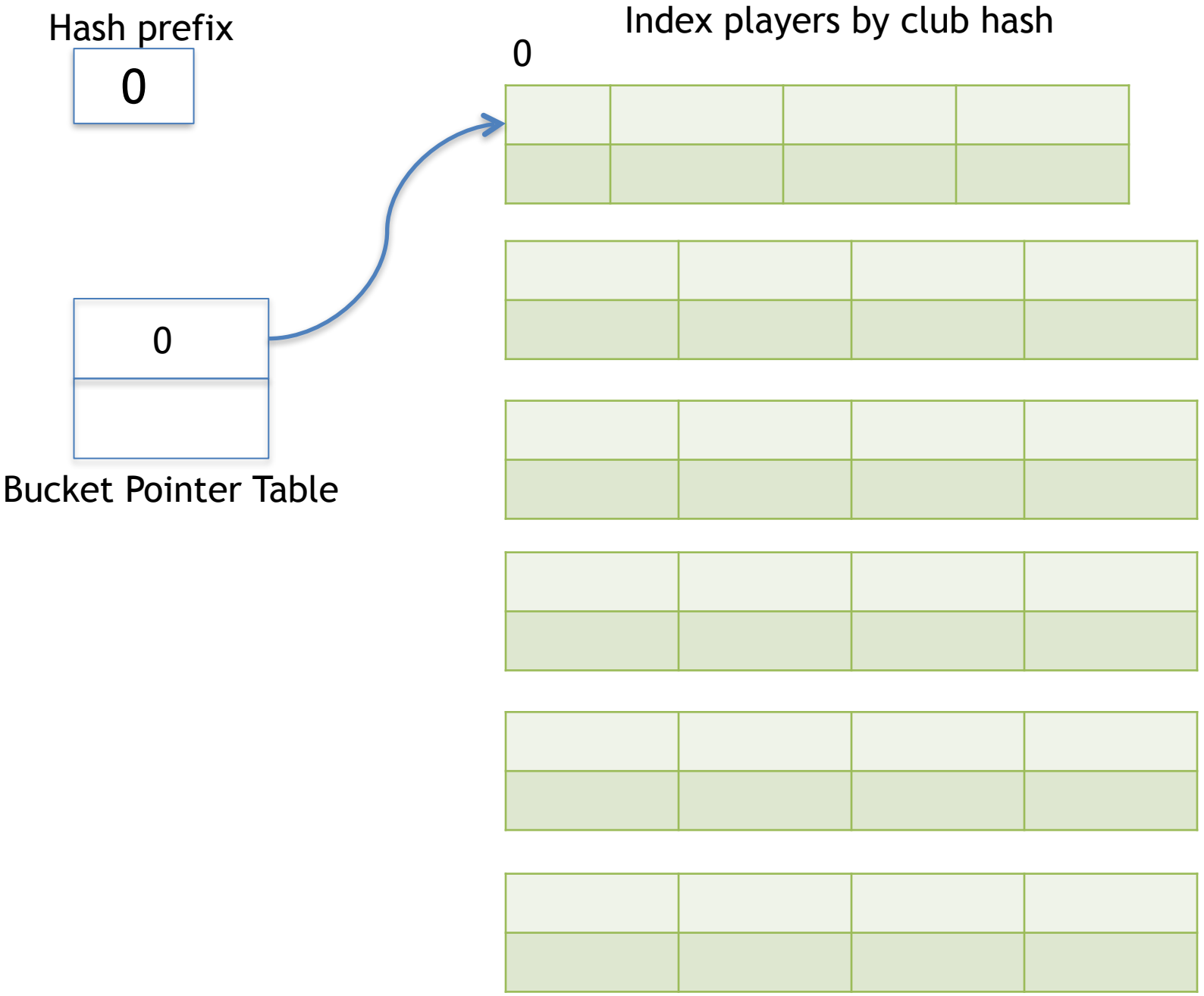
From Silberschatz, et.al., *Database Systems Concepts*, 6th ed., updated by CCPalmer

Players

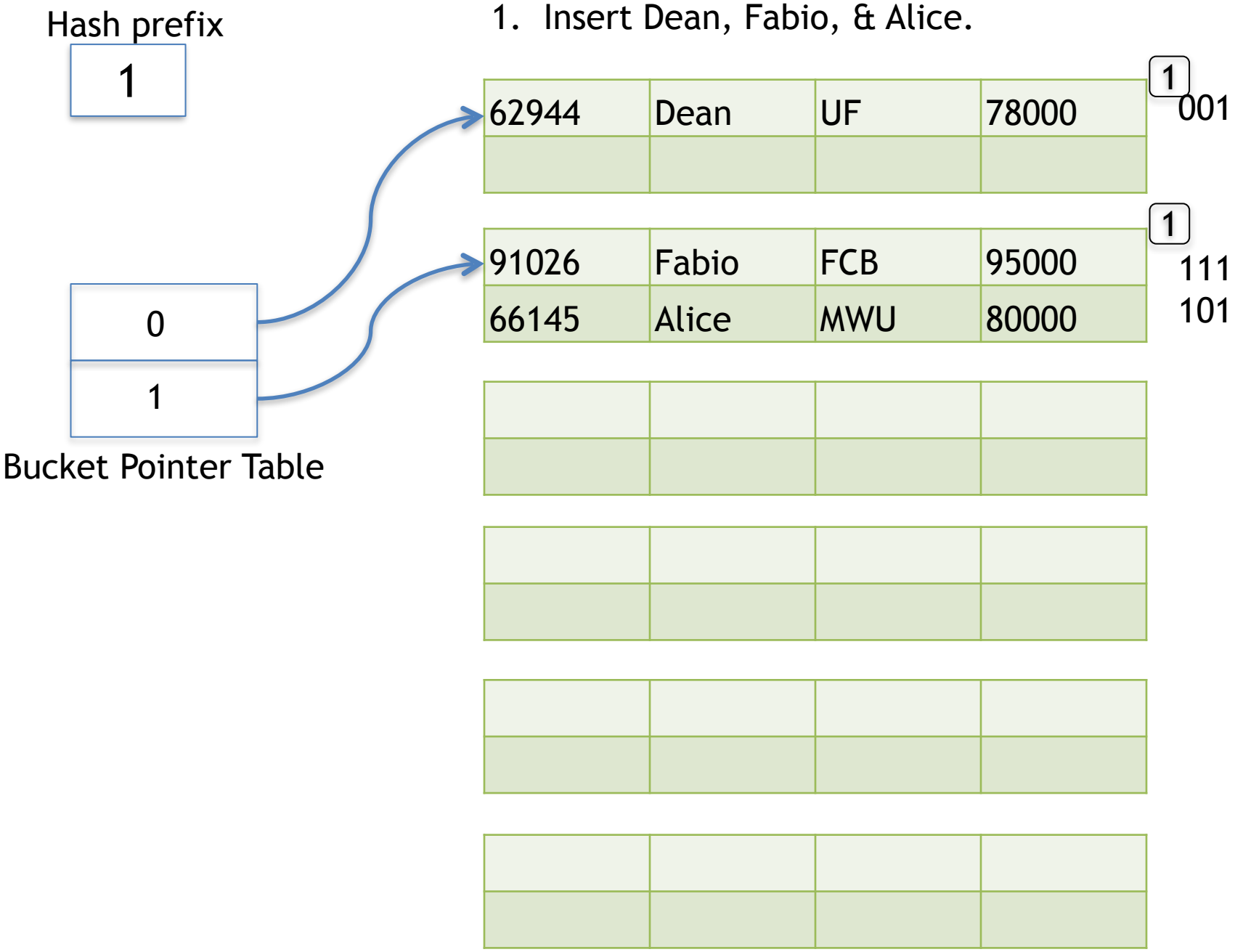
ID	Name	Club	Salary
66145	Alice	MWU	80000
35643	Boris	FCB	89000
49511	Carlos	MC	76000
62944	Dean	UF	78000
68288	Eve	MWU	93500
91026	Fabio	FCB	95000
14838	Gina	LFC	82000
53062	Hector	FCB	92000
74400	Ignatio	LFC	67700
76190	Jose	GFC	81800
84231	Kyle	GFC	77700
56190	Luis	CFC	82000

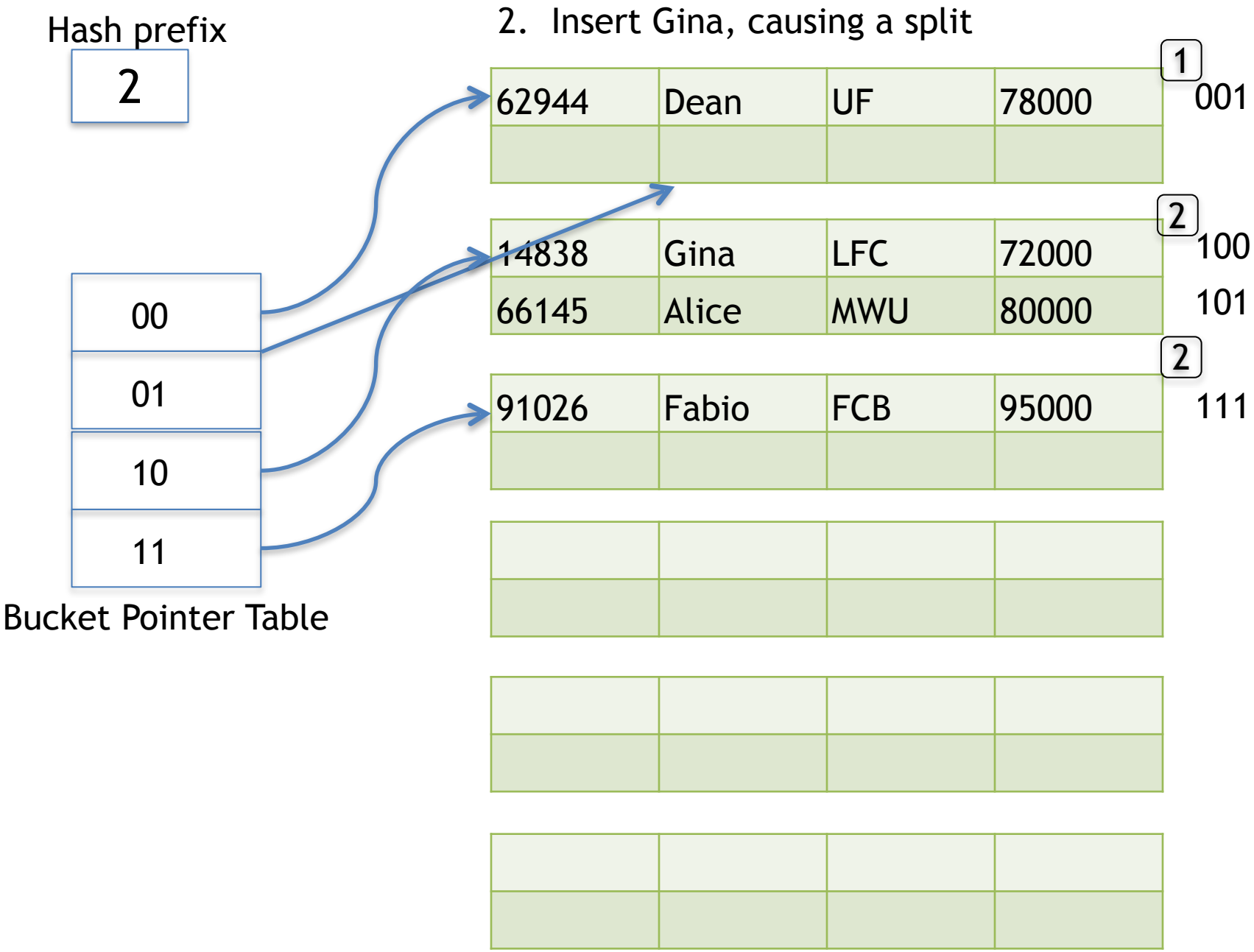
Hash Values

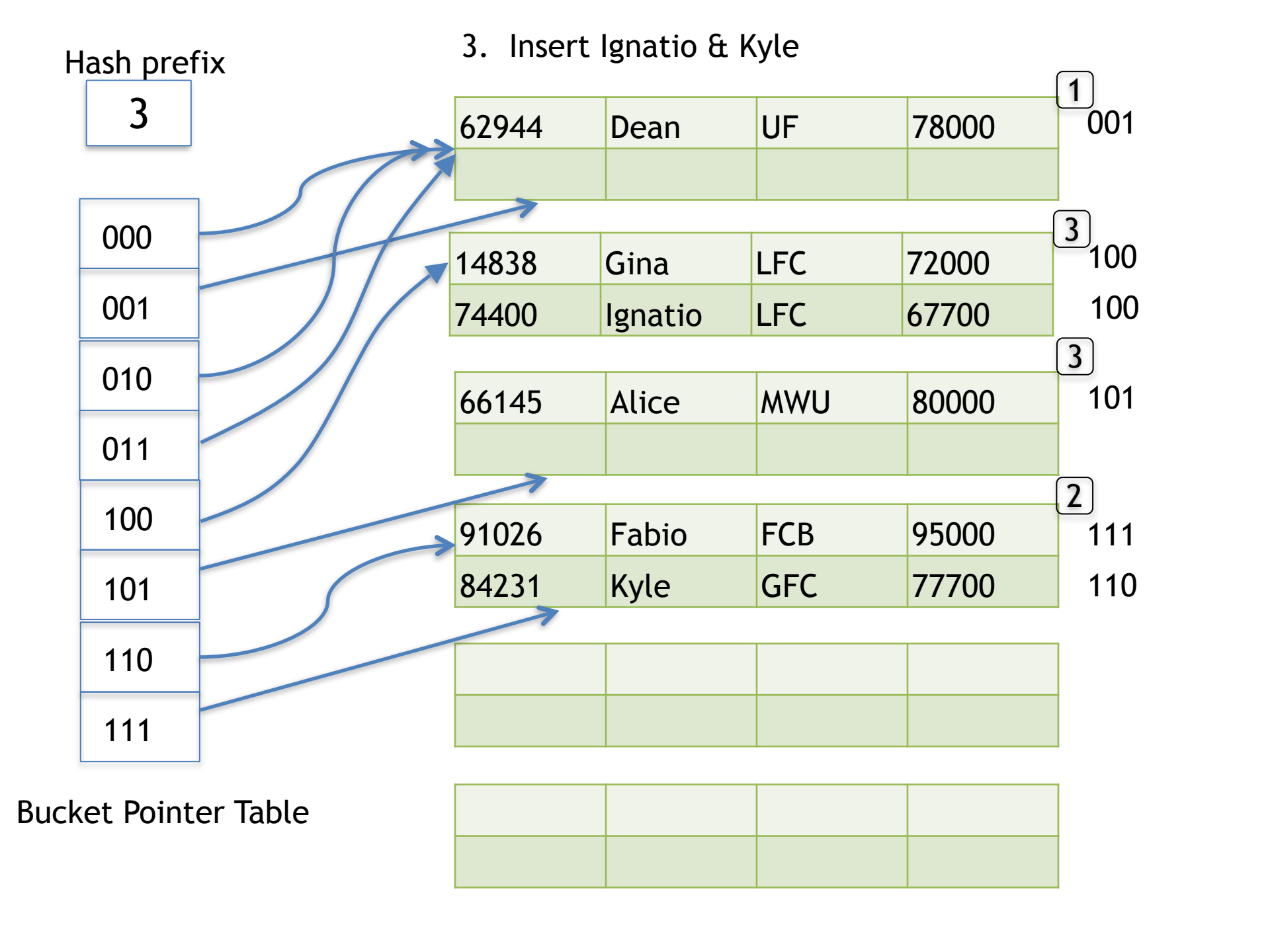
Club	**hash(Club)**		
CFC	**00111111**	**10101100**	**00010000**
FCB	**11111001**	**10010000**	**00100111**
GFC	**11000111**	**10010001**	**01111100**
LFC	**10010110**	**00000111**	**00100100**
MC	**01001101**	**11101110**	**00000101**
MWU	**10100010**	**11000010**	**01011111**
UF	**00110011**	**01110010**	**00001110**


Hash prefix
0
0
Bucket Pointer Table
Index players by club hash
0

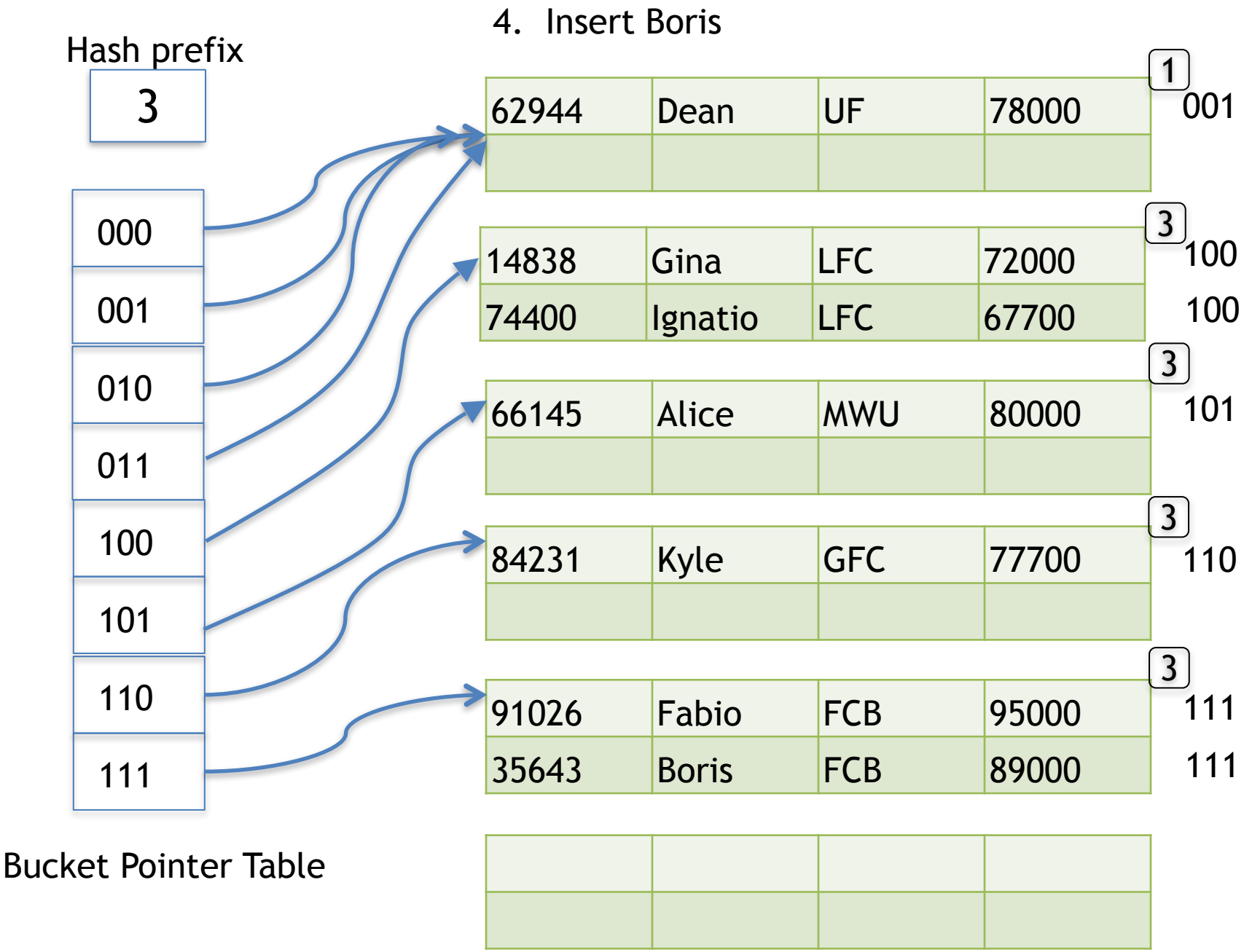
1. Insert Dean, Fabio, & Alice.



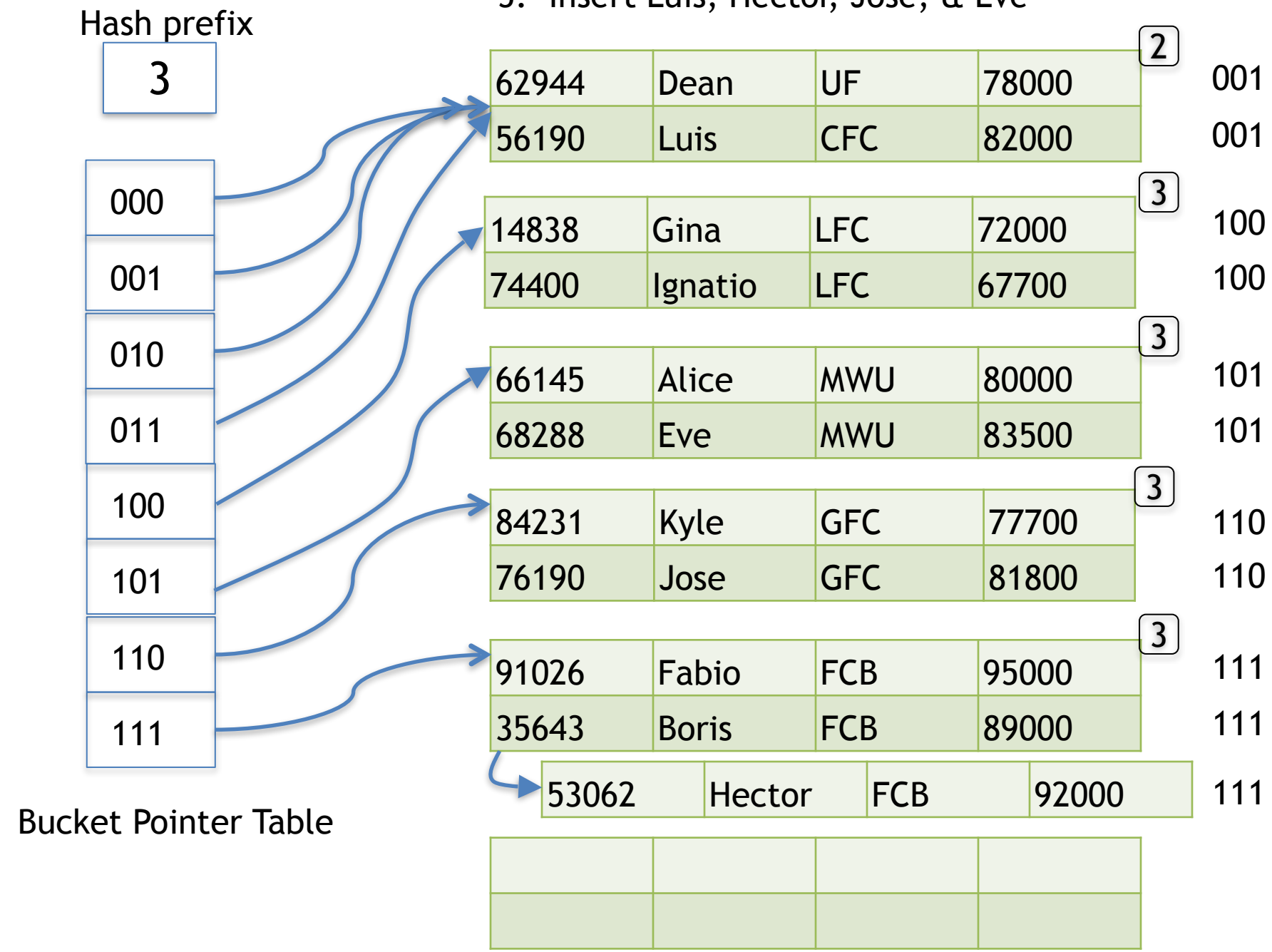

Hash prefix
2
00
01
10
11
Bucket Pointer Table
2. Insert Gina, causing a split
62944
Dean
UF
78000
1
001
14838
Gina
LFC
72000
2
100
66145
Alice
MWU
80000
101
91026
Fabio
FCB
95000
2
111


3. Insert Ignatio & Kyle
Hash prefix
3
000
001
010
011
100
101
110
111
Bucket Pointer Table
62944 Dean UF 78000
1
001
14838 Gina LFC 72000
74400 Ignatio LFC 67700
3
100
100
66145 Alice MWU 80000
3
101
91026 Fabio FCB 95000
84231 Kyle GFC 77700
2
111
110

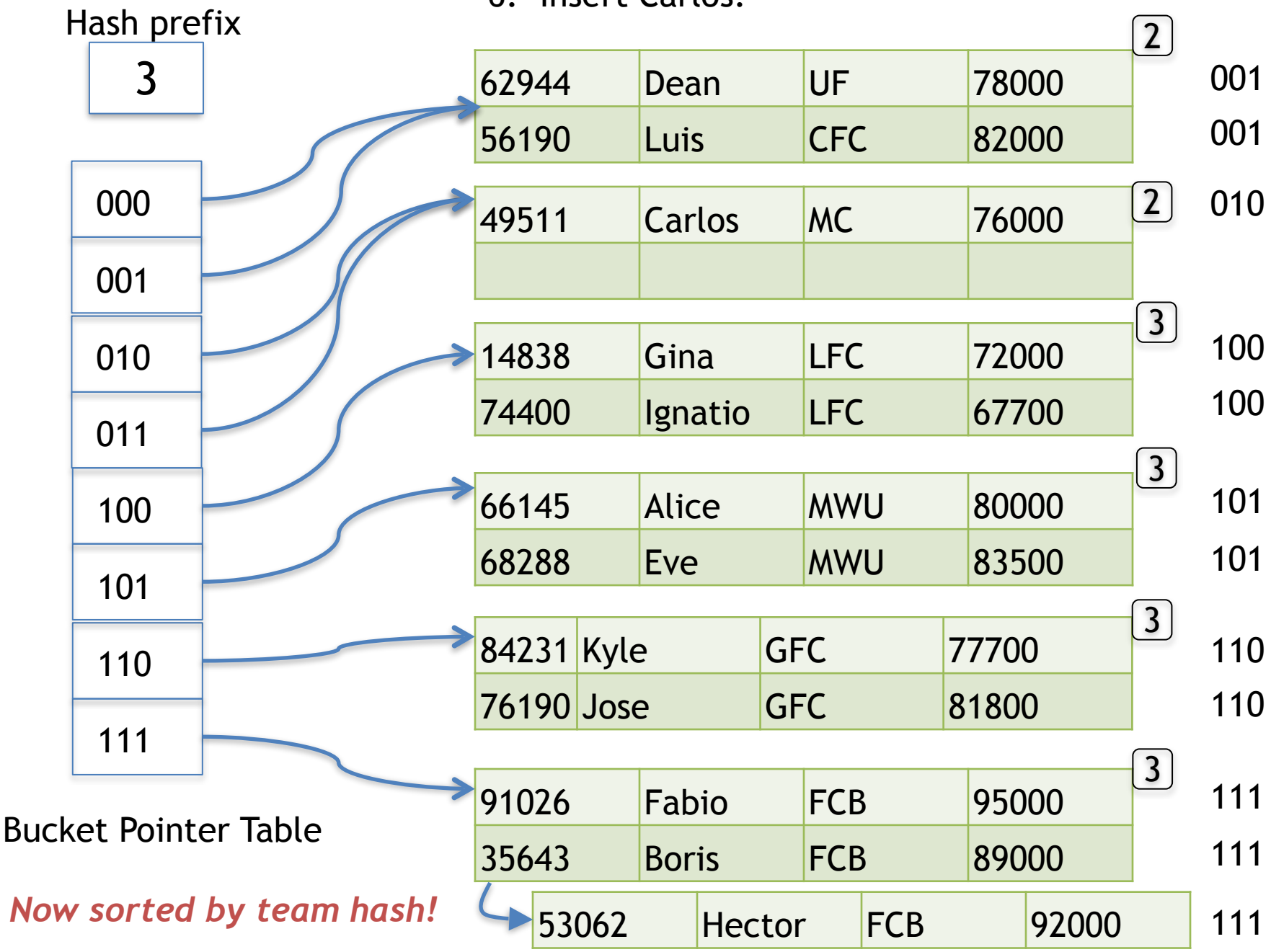
4. Insert Boris



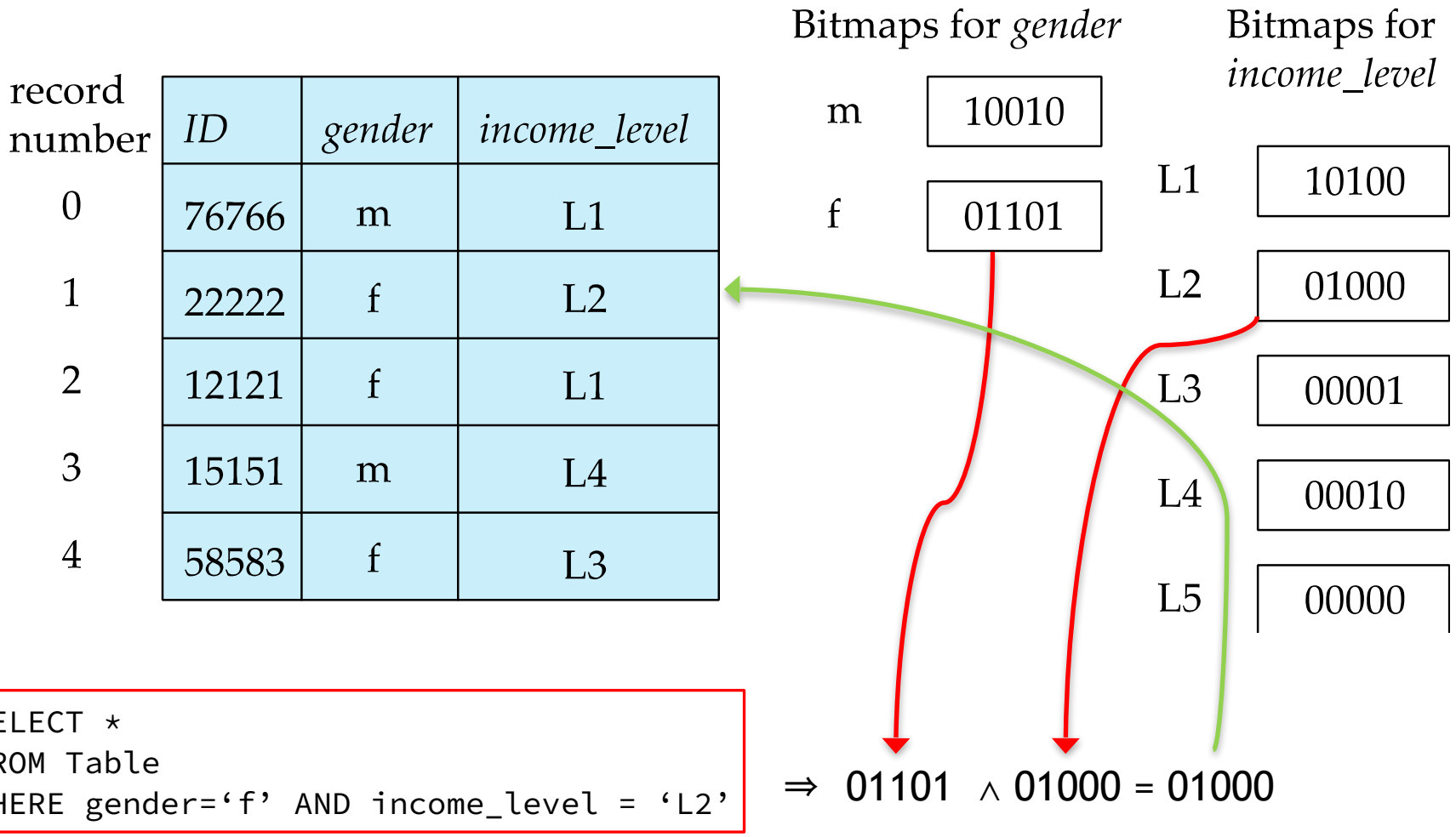
5. Insert Luis, Hector, Jose, & Eve



6. Insert Carlos.



Bitmap Indices




```
SELECT *
FROM Table
WHERE gender='f' AND income_level = 'L2'
```

Adapted from Silberschatz, et.al., *Database Systems Concepts*, 6th ed.

Bitmap Indices

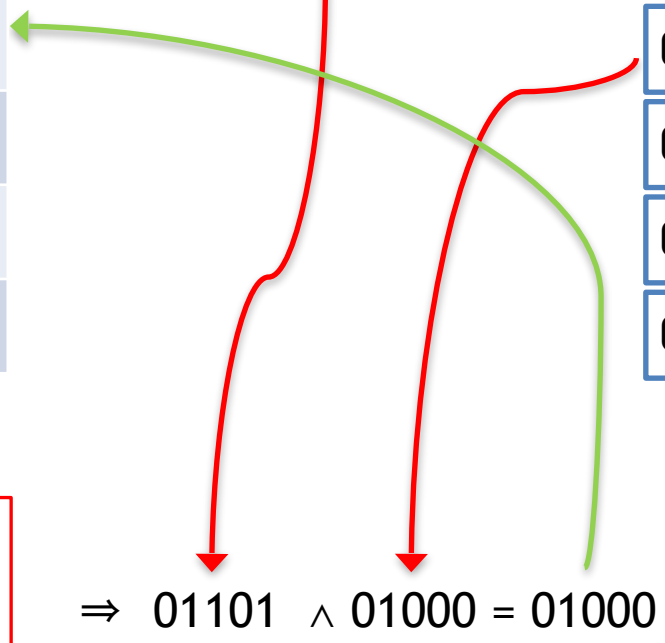
Record #	ID	Grad	Income Level
0	76766	Yes	L1
1	22223	No	L2
2	12641	No	L1
3	86538	Yes	L4
4	49650	No	L3

Bitmaps for *College grad*

Yes	10010
No	01101

Bitmaps for *Income Level*

10100
01000
00001
00010
00000



```
SELECT *
FROM Table
WHERE Grad="No" AND income_level = 'L2'
```

$\Rightarrow$ 01101 $\wedge$ 01000 = 01000

Adapted from Silberschatz, et.al., *Database Systems Concepts*, 6th ed.