ACS – 3902 - 002

Database Systems

Assignment 4:

Student#: 3129620

Student Name: Dai Dai Vo

Date: March 31, 2023

1.

EMPLOYEE:

Ssn -> Bdate

Ssn -> Fname

Ssn -> Minit

Ssn -> Lname

Ssn -> Address

Ssn -> Salary

Ssn -> Sex

Ssn -> DEPARTMENT.Number, DEPARTMENT.Name

DEPARTMENT:

Name, Number -> Number\_of\_employees

PROJECT:

Name, Number -> Location

Name, Number -> DEPARTMENT.Number, DEPARTMENT.Name

DEPENDENT:

EMPLOYEE.Ssn, Name -> Sex

EMPLOYEE.Ssn, Name -> Birth\_date

EMPLOYEE.Ssn, Name -> Relationship

MANAGES:

EMPLOYEE.Ssn, DEPARTMENT.Name, DEPARTMENT.Number -> Start\_date

WORKS\_ON:

EMPLOYEE.Ssn, PROJECT.Name, PROJECT.Number -> Hours

2.

a.

STUDENT:

|  |  |  |  |
| --- | --- | --- | --- |
| sNum | Name | Course | Hobby |
| 101 | Sam | Math, English, Art | Singing, Dancing |
| 102 | Bob | Math, Physics | Tennis, Soccer |
| 103 | Phillips | Biology, Chemistry | Photography, Cooking |
| 104 | David | Math, English | Hiking, Gaming |

b.

FDs:

sNum -> Name

MVDs:

sNum ->> Course

sNum ->> Hobby

c.

Student is in NF3 because there is no partial dependency and transitive dependency in table. However, there is still MVDs in the table, so it can not be NF4.

d.

ENROLLS:

|  |  |
| --- | --- |
| sNum | Course |
| 101 | Math |
| 101 | English |
| 101 | Art |
| 102 | Math |
| 102 | Physics |
| 103 | Biology |
| 103 | Chemistry |
| 104 | Math |
| 104 | English |

HOBBIES:

|  |  |
| --- | --- |
| sNum | Hobby |
| 101 | Singing |
| 101 | Dancing |
| 102 | Tennis |
| 102 | Soccer |
| 103 | Photography |
| 103 | Cooking |
| 104 | Hiking |
| 104 | Gaming |

STUDENTS:

|  |  |
| --- | --- |
| sNum | Name |
| 101 | Sam |
| 102 | Bob |
| 103 | Phillips |
| 104 | David |

3.

a.

Initially, there were two buckets and zero value with p = 0 and L = 0. After adding two values using (value mod 2) as the hash function, one more bucket was added to the table. After that, the values in p = 0 was redistributed using (value mod 4), then p = 1. When adding the next two value, if the result of the (value mod 2) was 0, which is smaller than p = 1, the value would have been hashed with value has 4. After adding two more value, one more bucket was added and there were four buckets, which was double the size of the initial hash table. When that happened, p was reset to 0 and L was increased by one, which caused the hash function for non splitting purpose or hash value < p to become (value mod 4). The pattern kept going on like that until we have eight value and four table were added so we now have six buckets with index from 0 to 5. The p travelled four times, went from 0 -> 1 -> 0 -> 1 -> 2.

16 is in bucket 0 since 16 mod 2 = 0, 16 mod 4 = 0, and 16 mod 8 = 0.

22, 30, 2 are in bucket 2 since they all have the result of 0 when mod 2 but when they mod 4, they result in 2. Therefore, they are the result of the split from bucket 0 when L = 0, or they were added after p = 1 and they mod 2 = 0 < p, or they were added after L = 1.

7, 27 are in bucket 3 since they all have the result of 1 when mod 2 but when they mod 4, the result is 3. Therefore, they are the result of the split from bucket 1 when L = 0, or they were added after L = 1.

13, 5 are in bucket 5 since they all have the result of 1 when mod 2, and mod 4. When they mod 8, the result is 5. Therefore, they are the result of the split form bucket 1 after L = 1.

b.

With L = 1, we have the hash function of: key mod (2 \* 21) = key mod 4.

The hash function for splitting will be: key mod (N \* 2L+1) = key mod (2 \* 21+1 )= key mod 8

To insert 18:

18 mod 4 = 2 which is = p. So, 18 will be added to bucket 2.

To insert 10:

10 mod 4 = 2 which is = p. So, 10 will be added to bucket 2.

Since adding 10 is the second insert, we do the split. Bucket 6 is added, and we split values in bucket 2 since p = 2.

22 mod 8 = 6, 30 mod 8 = 6, 2 mod 8 = 2, 18 mod 8 = 2, 10 mod 8 = 2. Therefore, 2, 18, 10 stay at the same bucket. 22 and 30 will be moved to bucket 6.

p = p + 1 = 3

To insert 35

35 mod 4 = 3, which is = p. So, 35 will be added to bucket 3.

To insert 12

12 mod 4 = 0, which is < p. So we use the next family of hash function, which is (key mod (N \* 2L+1) = (key mod (2 \* 21+1) = (key mod 8)

12 mod 8 = 4. So, 12 will be added to bucket 4.

Since adding 12 is the second insert, we do the split. Bucket 7 is added, and we split values in bucket 3 since p = 3.

7 mod 8 = 7, 27 mod 8 = 3, 35 mod 8 = 3. Therefore, 7 will be moved to bucket 7, whereas 8 and 3 will stay in bucket 3.

The hash table once again double in size because have eight buckets and the last double in size result in the hash table with four buckets. L = L + 1 = 1 + 1 = 2

p is reset to 0

From now we use the next family of hash function, which is (key mod (N \* 2L) = (key mod (2 \* 22) = (key mod 8)

With L = 2, we have the hash function of: key mod (2 \* 22) = key mod 8

To insert 74

74 mod 8 = 2. So, 74 will be added to bucket 2.

To insert 57

57 mod 8 = 1. So, 57 will be added to bucket 1.

Since adding 57 is the second insert, we do the split. Bucket 8 is added, and we split values in bucket 0 since p = 0.

16 mod 16 = 0. Therefore, 16 stays at bucket 0.

p = p + 1 = 1

To insert 8

8 mod 8 = 0, which is < p. So we use the next family of hash function, which is (key mod (N \* 2L+1) = (key mod (2 \* 22+1) = (key mod 16).

8 mod 16 = 8. So, 8 will be moved to bucket 8.

To insert 11

11 mod 8 = 3, which is > p. So, 11 will be added to bucket 3.

Since adding 11 is the second insert, we do the split. Bucket 9 is added, and we split values in bucket 1 since p = 0.

57 mod 16 = 9. So, 57 will be move to bucket 9

p = p + 1 = 2

4.

Since p = 4, internal node will have a minimum of 2 pointers and maximum of 4 pointers – inserting a fifth will cause a split. Leaf can have at least 2 key/pointer pair and a maximum of 4 key/pointers pair -inserting a fifth will cause a split.

Diagram

Description automatically generated

Add Dar

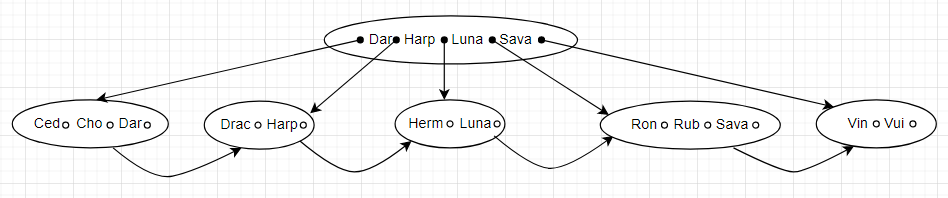
Dar will be added before Drac and after Cho, and since p = 4, the leaf of Ced, Cho, Dar, Drac, Harp will be split into one leaf of Ced, Cho, Dar, and one leaf of Drac and Harp. Dar will also be moved up to the upper internal node.

Diagram

Description automatically generated

Add Vui

Vui will be added after Vin, and since p = 4, the leaf of Ron, Rub, Sava, Vin, Vui will be split into one leaf of Ron, Rub, Sava, and one leaf of Vin and Vui. Sava will also be moved up to the upper internal node.



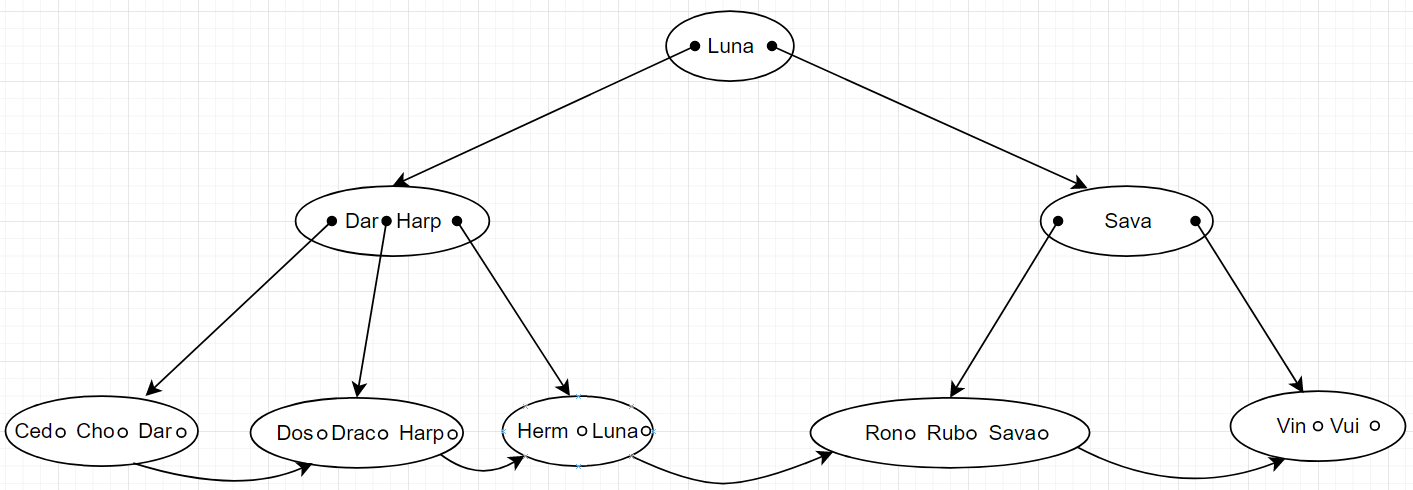
But since p = 4, one internal node can only have the most four pointers. Because of that, the internal node above will split

Diagram

Description automatically generated

Add Dos

Dos will be added after Dar



Add Mal

Dos will be added before Ron

Diagram

Description automatically generated

Add Row

Row will be added before Rub and after Ron and since p = 4, the leaf of Mal Ron, Row, Rub, and Sava will be split into one leaf of Mal, Ron, Row and one leaf of Rub and Sava. Row will also be moved up to the upper internal node.

Diagram

Description automatically generated with low confidence

Add Jos

Dos will be added before Luna and after Herm

Diagram

Description automatically generated with medium confidence

Add Lam

Lam will be added before Luna and after Jos

Diagram

Description automatically generated

Add Zin

Zin will be added after Vui

Diagram

Description automatically generated with medium confidence