ICCS313: Assignment 5

Phairat Lin
phairat.lin@student.mahidol.edu
Collaborators: All my friends in 1409

2: Relational Database Design Theory

---

(1)
Schema:

```
Contracts(c_no, supp_no, proj_no, dept_no, part_no, qty, val)
```

Alice's findings:

```
c_no -> supp_no proj_no dept_no part_no qty val
proj_no part_no -> c_no
supp_no dept_no -> part_no
```

Bob's findings:

```
c_no -> proj_no supp_no dept_no
proj_no part_no -> c_no
supp_no proj_no dept_no -> c_no part_no qty val
```

No, they are not equivalent. In Alice's findings, You can access part_no with just supp_no and dept_no, but in Bob's findings, you cannot. You need proj_no in addition to supp_no and dept_no to access part_no.
There is no other functional dependency to access part_no without using proj_no, which is different from Alice's findings.

(2)
Let us assume R1 as Table 1 and R2 as table 2.

α_1	β
A1	B1
A2	B2

Table 1: R_1 = (α_1, β)

α_1	γ
A1	Y1
A2	Y2

Table 2: R_2 = (α_1, γ)

For the LHS, let us assume R as Table 3.

α_1	β	γ
A1	B1	Y1
A2	B2	Y2

Table 3: (α_1, β, γ)

For the RHS, $R1 \bowtie R2 = \Pi_{A \cup B}(\sigma_{\alpha1=\alpha2}(\rho_{\alpha1 \to \alpha2}(R1) X R2)) \longrightarrow 1$, as shown in Table 4, Table 5, and Table 6.

α_1	β	α_2	γ
A1	B1	A1	Y1
A2	B2	A2	Y2
A1	B1	A2	Y1
A2	B2	A1	Y2

Table 4: $\rho_{\alpha_1 \to \alpha_2}$(R1) X R2

α_1	β	α_2	γ
A1	B1	A1	Y1
A2	B2	A2	Y2

Table 5: $(\sigma_{\alpha_1=\alpha_2}(\rho_{\alpha_1 \to \alpha_2}(R1) X R2))$

In Table 6, the union will remove redundancy.

α_1	β	γ
A1	B1	Y1
A2	B2	Y2

Table 6: $\Pi_{A \cup B}(\sigma_{\alpha_1=\alpha_2}(\rho_{\alpha_1 \to \alpha_2}(R1) X R2))$

From this, we can see that Table 6 is equivalent to Table 3 which means $R = R1 \bowtie R2$.

(3)

- No, R is not in BCNF because the determinant CD is not a candidate key.
- Decomposing R will give us R_1 and R_2. R_1 will contain BCD and R_2 will contain CDA.
  $R_1(BCD)$

  ```
  Key: BC
  BC → D
  ```

  $R_2(CDA)$

  ```
  Key: CD
  CD → A
  ```

- No it is not lossless.
  If it is lossless, then one of the following will occur:

$$R_1 \cap R_2 \to R_1$$

$$R_1 \cap R_2 \to R_2$$

  $R_1 \cap R_2 = AC$
  We can check if $AC \to BCD$ or $AC \to CDA$ by looking at the closures of all attributes.

$$AB = \{A, B, C, D\}$$

$$BC = \{A, B, C, D\}$$

$$CD = \{A, C, D\}$$

  From the above, we can see clearly that $AC \to BCD$ or $AC \to CDA$ is not in the closures, hence it is not lossless.

- Yes R is in 3NF because even though CD is not a candidate key, A is in the candidate keys which means it is not prime.

3: Storage and Indexing

---

(1)

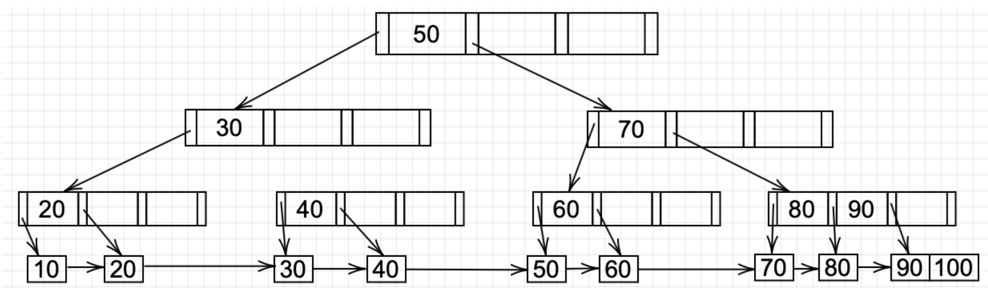
- Find all tuples of R.
  The third approach, scanning through the whole heap file, would most likely require the fewest I/O operations

- Find all R tuples such that R.A $\in$ [0, 100)
  The first approach, using B^+ tree index on R.A., would most likely require the fewest I/O operations due to properties like balanced binary tree.

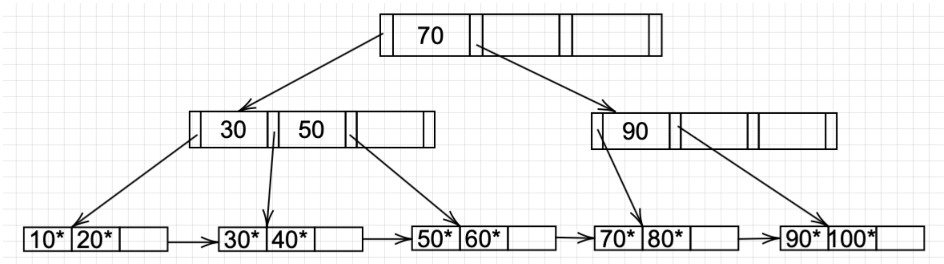
- Find all R tuples such that R.A = 100
  The second approach, using hash index on attribute R.A., would most likely require the fewest I/O operations since we obtain the tuples right away, costing $O(1)$ at most.

(2)

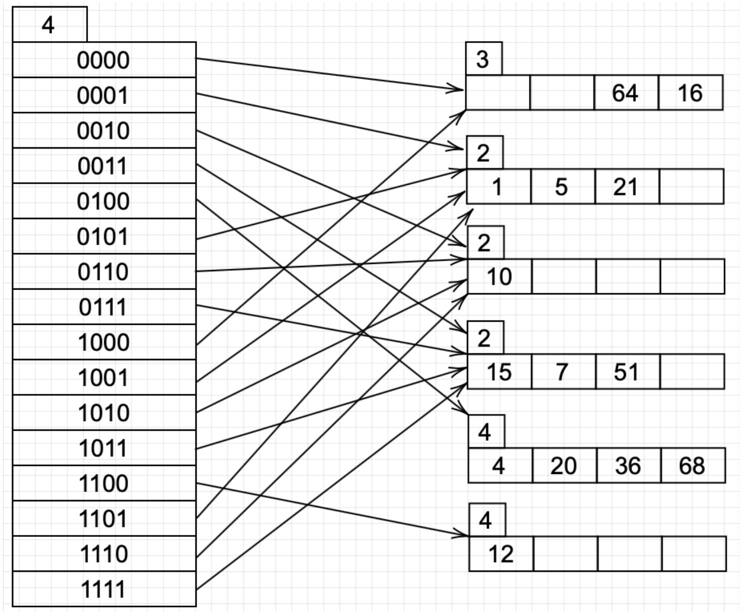
- Below is the tree with keys inserted in ascending order.



- Yes it is possible by inserting keys in descending order as shown below.



(3)
(a)



(b)
The minimal set of entries to be deleted from the index that would trigger a merge is {16, 64} since it'll result in an empty bucket.

4: Exam revisited

---

(1) Set-model will show distinct results, therefore students with the same score will not be shown.

(2)

```
SELECT distinct maker
FROM Computer
INNER JOIN PC on Computer.model = PC.model
WHERE speed >= ALL
(
SELECT speed
FROM PC
)
```

(3) The candidate keys are BD and CD because they hold relations to every attribute.

(4)
- A minimal basis must have single element on the RHS. Since $\mathcal{G}$ has more than one element on the RHS, $\mathcal{G}$ is not a minimal basis of $\mathcal{F}$.
- Since $A \to B$ and $A \to C$ can be combined into $A \to BC$ and $A \to BC$ can be split into $A \to B$ and $A \to C$, $\mathcal{G}$ is a basis of $\mathcal{F}$ and vice versa.
- The minimal basis of $\mathcal{F}$ is $\{A \to B, A \to C\}$