Final Review

CS157A

Normalization

- Suppose we have a database consisting of the following attributes: R(B, O, I, S, Q, D)
- B – Broker, O – Office of a broker, I – Investor, S – Stock
- Q – Quantity of stock owned by an investor, D – Dividend paid by a stock.
- Assume that the following FDs: I → B, IS → Q, B → O, S → D.
- Find a key for the relation schema R = (B, O, I, S, Q, D).
  - ***I and S must be in any candidate key*** since they do not appear on the right of any FD. The question is whether they form a complete candidate key. And yes, IS → ISDBOQ. Hence, the only candidate key is IS.

Normalization

- Consider a relation R with attributes A, B, C, D, E and the set of functional dependencies A⟶CD, B⟶CE, E⟶B. We decompose into R1 = (A, B, E), R2 = (A, C, D). Show that this decomposition is a lossless-join decomposition.
- R1∩ R2 = A; and A is a Key in R2. this is a lossless-join decomposition
- Show that this decomposition is not a dependency-preserving decomposition.
- F1 = {E → B} F2 = {A → CD}
- (F1 ∪ F2)+≠ F+
- Not a dependency preserving decomposition. B->CE is lost.

Schema Refinement

- FD Equivalency
- Consider F={A → B, B → C, C → A} and G={C → B, B → A, A → C}
- Are F and G equivalent?
- If F covers G
  - All FDs in G can be derived from F

  G: C → B B → A A → C

  $C_F^+ = CA\underline{B}$ $B_F^+ = BC\underline{A}$ $A_F^+ = AB\underline{C}$
- If G covers F
  - All FDs in F can be derived from G

  F: A → B B → C C → A

  $A_G^+ = AC\underline{B}$ $B_G^+ = BA\underline{C}$ $C_G^+ = CB\underline{A}$
- Therefore, F ≡ G

Schema Refinement

- Minimal Cover
- Consider F={A → D, BC → AD, C → B, E → A, E → D}
- Find Minimal cover of F?
- All FDs have singleton RHS
  - BC → AD: Decompose to BC → A, BC → D
  - F={A → D, BC → A, BC → D, C → B, E → A, E → D}
- Find extraneous attribute in LHS
  - Examine LHS of each FD that has 2 or more attributes.
  - BC → A
    - $B^+ = B$ $C^+ = CBAD$ (eliminate B)
  - BC → D
    - $B^+ = B$ $C^+ = CBAD$ (eliminate B)
  - F ={A → D, C → A, C → D, C → B, E → A, E → D}

Schema Refinement

- F ={A → D, C → A, C → D, C → B, E → A, E → D}
- Eliminate redundant FDs
  - Examine each FD one at a time, see if FD is needed, if not eliminate
  - A → D
    - Pretend this FD does not exist and see if we can determine D from A
    - A^+ = A (Keep)
  - C → A
    - C^+ = CDB (Keep)
  - C → D
    - C^+ = CADB (includes D, eliminate)
  - C → B
    - C^+ = CAD (Keep)
  - E → A
    - E^+ = ED (Keep)
  - E → D
    - E^+ = EAD (eliminate)

F = {A → D, C → A, C → B, E → A}

Schema Refinement

- Minimal Cover
- Consider F={A → G, F → DE, G → DB, D → C, E → D}
- Find Minimal cover of F?
- All FDs have singleton RHS
  - F → DE: Decompose to F → D, F → E
  - G → DB: Decompose to G → D, G → B
  - F={A → G, F → D, F → E, G → D, G → B, D → C, E → D}
- Find extraneous attribute in LHS
  - None – No LHS with 2 or more attributes

Schema Refinement

- F={A → G, F → D, F → E, G → D, G → B, D → C, E → D}
- Eliminate redundant FDs
  - A → G
    - A^+ = A (Keep)
  - F → D
    - F^+ = FED (eliminate)
  - F → E
    - F^+ = F (Keep)
  - G → D
    - G^+ = GB (Keep)
  - G → B
    - G^+ = GDC (Keep)
  - D → C
    - D^+ = D (Keep)
  - E → D
    - E^+ = E (Keep)

F={A → G, F → E, G → D, G → B, D → C, E → D}

Schema Refinement

- Minimal Cover
- Consider F={A⟶BC, B⟶C, A⟶B, AB⟶C}
- Find Minimal cover of F?
- All FDs have singleton RHS
  - A⟶BC: Decompose to A⟶B, A⟶C
  - F={A⟶B, A⟶C, B⟶C, AB⟶C}
- Find extraneous attribute in LHS
  - AB⟶C
    - $A^+ = ABC$
    - $B^+ = BC$ (both A and B are extra)
  - F={A⟶B, A⟶C, B⟶C}

Schema Refinement

- F={A → B, A → C, B → C}
- Eliminate redundant FDs
  - A → B
    - A^+ = AC (Keep)
  - A → C
    - A^+ = ABC (eliminate)
  - B → C
    - B^+ = B (Keep)

F={A → B, B → C}

Schema Refinement

- R(A, B, C, D, E)
- Consider F={A → BC, CD → E, B → D, E → A}
- List all Candidate Keys of R?
  - A^+ = ABCDE
  - B^+ = BD
  - CD^+ = ABCDE
  - E^+ = ABCDE

CK={A, CD, E}

Normalization

- R(A, B, C, D, E, F)
- Consider F={BC → A, AC → DE, F → E}
  - BCF^+ = ABCDEF

  CK={BCF}
- Is R in 1NF? Yes
- Is R in 2NF? No
  - R is in 1NF
  - Every non-candidate key attribute is fully functionally dependent on a candidate key (no partial dependency)
- Is R in 3NF? No
- Is R in BCNF? No

Schema Refinement

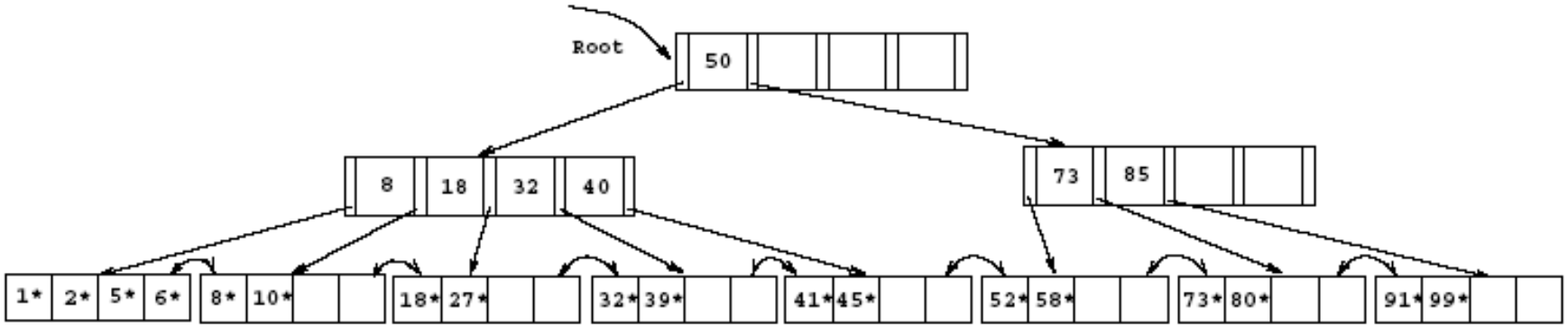
- R(A, B, C, D, E)
- Consider F={ABC → DE, D → A}
- List all Candidate Keys?
  - $BC^+ = BC$
  - $ABC^+ = ABCDE$
  - $DBC^+ = ABCDE$
  - $EBC^+ = EBC$ CK={ABC, DBC}
- Do a loss-less join decomposition into BCNF that preserves dependency or show such decomposition not possible?
  - ABC → DE (LHS contains KEY – OK)
  - D → A (LHS dosr NOT contain KEY – decompose)
    - R-Y: (BCDE) XY: (DA)
    - ABC → DE Lost in R

Normalization

- R(A, B, C, D)
- Consider F={AB → C, C → D}
- List all Candidate Keys?
  - AB^+ = ABCD CK={AB}
- Is R in 3NF?
  - First check if R in 2NF? Yes
    - R in 1NF
    - Every non-candidate key attribute is fully functionally dependent on a candidate Key (no partial dependency)
  - If R is in 3NF? No C → D (decompose)
    - R-Y: (ABC)
    - XY: (CD)

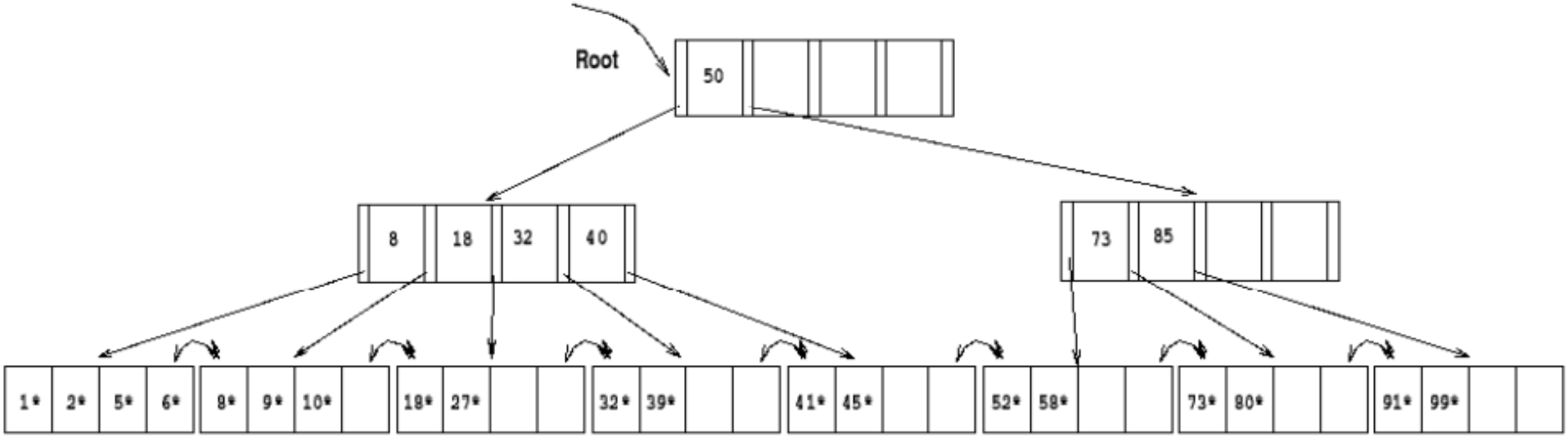
Indexing

- Consider the B+ tree index shown below



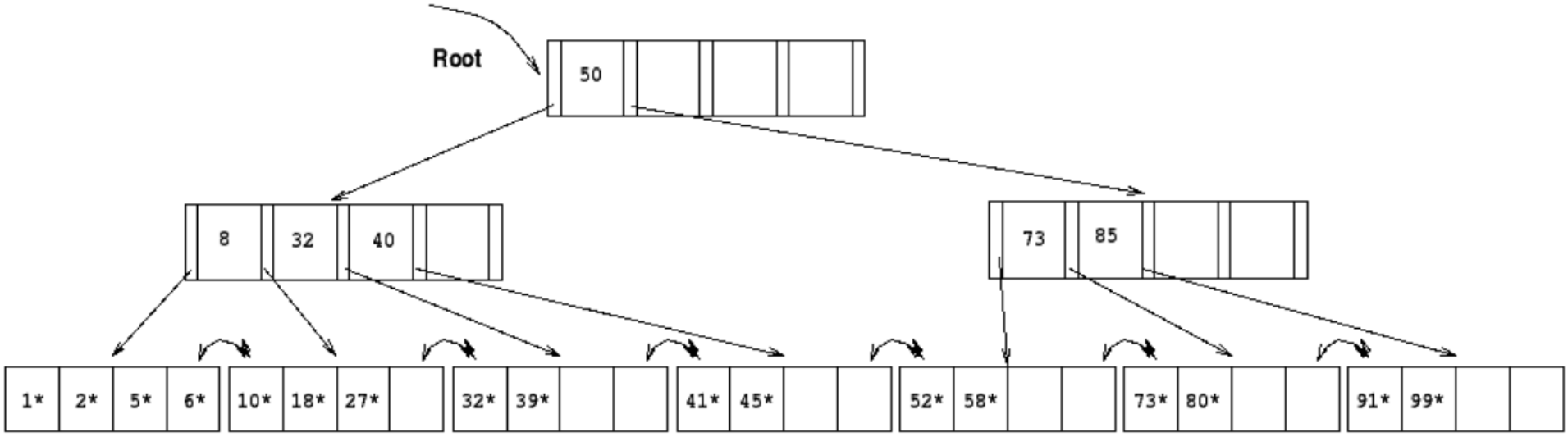
Indexing

- Show the tree that would result from inserting a data entry with key 9 into this tree.



Indexing

- Show the B+ tree that would result from deleting the data entry with key 8 from the original tree.



XML

- Consider the following XML DTD

```
<!ELEMENT bib (entry*)>
<!ELEMENT entry (book | collection)>
<!ELEMENT book (author, title)>
<!ELEMENT collection (author, title, publisher?, article+)>
<!ELEMENT article (author, title)>
<!ATTLIST entry (year CDATA #REQUIRED)>
<!ELEMENT author (#PCDATA)>
 <!ELEMENT title (#PCDATA)
```

XML

- Does the following XML fragment conform to the DTD above?

```
<bib>
    <entry>
        <year>2008</year>
            <collection>
                <author>A. Turing</author>
                <title>Tests in CS</title>
            </collection>
    <book>
        <title>The Hunger Games</title>
        <author>Suzanne Collins</author>
    </book>
</bib>
```

- No, because:
  - year is an attribute, not an element
  - Collection must have at least one article (but publisher can be missing)
  - Author must precede title in book
  - There is no closing “entry” tag

XML

Write down the XML schema for the following XML document?

```
 1: <?xml version="1.0"?>
 2: <BookStore xmlns="http://www.books.org"
 3:            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 4:            xsi:schemaLocation=
 5:                          "http://www.books.org
 6:                           BookStore.xsd">
 7:         <Book>
 8:                 <Title>My Life and Times</Title>
 9:                 <Author>Paul McCartney</Author>
10:                 <Date>1998</Date>
11:                 <ISBN>1-56592-235-2</ISBN>
12:                 <Publisher>McMillin Publishing</Publisher>
13:         </Book>
14:         <Book>
15:                 <Title>Illusions The Adventures of a Reluctant Messiah</Title>
16:                 <Author>Richard Bach</Author>
17:                 <Date>1977</Date>
18:                 <ISBN>0-440-34319-4</ISBN>
19:                 <Publisher>Dell Publishing Co.</Publisher>
20:         </Book>
21:         <Book>
22:                 <Title>The First and Last Freedom</Title>
23:                 <Author>J. Krishnamurti</Author>
24:                 <Date>1954</Date>
25:                 <ISBN>0-06-064831-7</ISBN>
26:                 <Publisher>Harper &amp;amp; Row</Publisher>
27:         </Book>
28: </BookStore>
```

XML

```
 1: <?xml version="1.0" encoding="UTF-8"?>
 2: <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 3:             targetNamespace="http://www.books.org"
 4:             xmlns="http://www.books.org"
 5:             elementFormDefault="qualified">
 6:     <xsd:element name="BookStore">
 7:         <xsd:complexType>
 8:             <xsd:sequence>
 9:                 <xsd:element ref="Book" minOccurs="1" maxOccurs="unbounded"/>
10:             </xsd:sequence>
11:         </xsd:complexType>
12:     </xsd:element>
13:     <xsd:element name="Book">
14:         <xsd:complexType>
15:             <xsd:sequence>
16:                 <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
17:                 <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
18:                 <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
19:                 <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
20:                 <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
21:             </xsd:sequence>
22:         </xsd:complexType>
23:     </xsd:element>
24:     <xsd:element name="Title" type="xsd:string"/>
25:     <xsd:element name="Author" type="xsd:string"/>
26:     <xsd:element name="Date" type="xsd:string"/>
27:     <xsd:element name="ISBN" type="xsd:string"/>
28:     <xsd:element name="Publisher" type="xsd:string"/>
29: </xsd:schema>
30:
```

Transactions

Consider the following two transactions and schedule (time goes from left to right). Is this schedule conflict-serializable? Explain why or why not.

T1: R1(A), W1(A), R1(B), W1(B), C
T2: R2(A), R2(B), C

The schedule is not conflict-serializable because the precedence graph contains a cycle.