TEST PROGRAM Graph Operations

**Graph operation use cases**

1. Add leaf node

3. Move leaf node

4. Insert node

6. Move subtree

2. Delete leaf node

5. Remove node

7. Delete subtree

**Atomic operations**

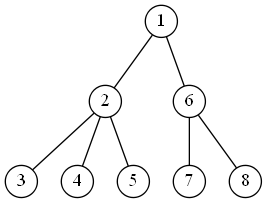
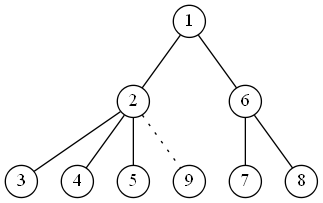
1. **Add** (leaf node)

2. **Delete** (leaf node)

3. **Move** (subtree)

Use cases decomposed to atomic operations

|  |  |  |
| --- | --- | --- |
| **#** | **Use case** | **Atomic operations** |
| 1 | Add leaf node | **Add** |
| 2 | Delete leaf node | **Delete** |
| 3 | Move leaf node | **Delete + Add** |
| 4 | Insert node | **Add + Move** |
| 5 | Remove node | **Move + Delete** |
| 6 | Move subtree | **Move** |
| 7 | Delete subtree | **Part of move** |

**1. Add leaf node**

Source graph Resulting graph

t\_data t\_path t\_data t\_path

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **node** | **parent** |  | **node** | **ancestor** |  | **node** | **ancestor** |  | **node** | **ancestor** |
| 1 | 1 |  | 2 | 1 |  | 1 | 1 |  | 2 | 1 |
| 2 | 1 |  | 3 | 2 |  | 2 | 1 |  | 3 | 2 |
| 3 | 2 |  | 3 | 1 |  | 3 | 2 |  | 3 | 1 |
| 4 | 2 |  | 4 | 2 |  | 4 | 2 |  | 4 | 2 |
| 5 | 2 |  | 4 | 1 |  | 5 | 2 |  | 4 | 1 |
| 6 | 1 |  | 5 | 2 |  | 6 | 1 |  | 5 | 2 |
| 7 | 6 |  | 5 | 1 |  | 7 | 6 |  | 5 | 1 |
| 8 | 6 |  | 6 | 1 |  | 8 | 6 |  | 6 | 1 |
|  |  |  | 7 | 6 |  | **9** | **2** |  | 7 | 6 |
|  |  |  | 7 | 1 |  |  |  |  | 7 | 1 |
|  |  |  | 8 | 6 |  |  |  |  | 8 | 6 |
|  |  |  | 8 | 1 |  |  |  |  | 8 | 1 |
|  |  |  |  |  |  |  |  |  | **9** | **2** |
|  |  |  |  |  |  |  |  |  | **9** | **1** |

**Python testing fragments**

source tuples

t\_dataOutput = [{'node': 1, 'parent': 1},

{'node': 2, 'parent': 1},

{'node': 3, 'parent': 2},

{'node': 4, 'parent': 2},

{'node': 5, 'parent': 2},

{'node': 6, 'parent': 1},

{'node': 7, 'parent': 6},

{'node': 8, 'parent': 6}]

t\_pathOutput = [(2, 1),

(3, 2),

(3, 1),

(4, 2),

(4, 1),

(5, 2),

(5, 1),

(6, 1),

(7, 6),

(7, 1),

(8, 6),

(8, 1)]

resulting tuples

t\_dataOutput = [{'node': 1, 'parent': 1},

{'node': 2, 'parent': 1},

{'node': 3, 'parent': 2},

{'node': 4, 'parent': 2},

{'node': 5, 'parent': 2},

{'node': 6, 'parent': 1},

{'node': 7, 'parent': 6},

{'node': 8, 'parent': 6},

{'node': 9, 'parent': 2}]

t\_pathOutput = [(2, 1),

(3, 2),

(3, 1),

(4, 2),

(4, 1),

(5, 2),

(5, 1),

(6, 1),

(7, 6),

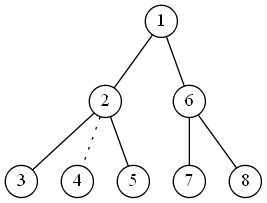
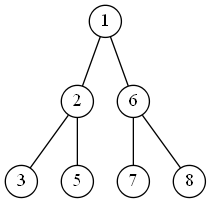
(7, 1),

(8, 6),

(8, 1),

(9, 2),

(9, 1)]

**2. Delete leaf node**

Source graph Resulting graph

t\_data t\_path t\_data t\_path

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **node** | **parent** |  | **node** | **ancestor** |  | **node** | **ancestor** |  | **node** | **ancestor** |
| 1 | 1 |  | 2 | 1 |  | 1 | 1 |  | 2 | 1 |
| 2 | 1 |  | 3 | 2 |  | 2 | 1 |  | 3 | 2 |
| 3 | 2 |  | 3 | 1 |  | 3 | 2 |  | 3 | 1 |
| **4** | **2** |  | **4** | **2** |  | 5 | 2 |  | 5 | 2 |
| 5 | 2 |  | **4** | **1** |  | 6 | 1 |  | 5 | 1 |
| 6 | 1 |  | 5 | 2 |  | 7 | 6 |  | 6 | 1 |
| 7 | 6 |  | 5 | 1 |  | 8 | 6 |  | 7 | 6 |
| 8 | 6 |  | 6 | 1 |  |  |  |  | 7 | 1 |
|  |  |  | 7 | 6 |  |  |  |  | 8 | 6 |
|  |  |  | 7 | 1 |  |  |  |  | 8 | 1 |
|  |  |  | 8 | 6 |  |  |  |  |  |  |
|  |  |  | 8 | 1 |  |  |  |  |  |  |

We need to delete the set:

*x* ⨯ *path*(*x*, *y*)

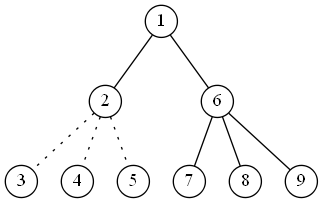
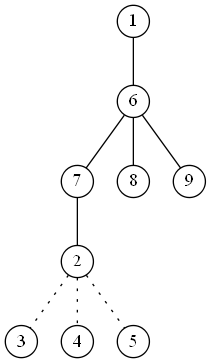
*x* = 4

*path*(4) = (2, 1)

*x* ⨯ *path*(4) = 4 ⨯ (2, 1) = ((4, 2), (4, 1))

**SQL construction**

DELETE FROM *t\_path* WHERE *node = x*

**6. Move subtree**

Looks like we need parent-to-root order, due to residue from deletion

Source graph Resulting graph

t\_data t\_path t\_data t\_path

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **node** | **parent** |  | **node** | **ancestor** |  | **node** | **ancestor** |  | **node** | **ancestor** |
| 1 | 1 |  | **2** | **1** |  | 1 | 1 |  | 3 | 2 |
| 2 | 1 |  | **3** | **1** |  | **2** | **7** |  | 4 | 2 |
| 3 | 2 |  | 3 | 2 |  | 3 | 2 |  | 5 | 2 |
| 4 | 2 |  | **4** | **1** |  | 4 | 2 |  | 6 | 1 |
| 5 | 2 |  | 4 | 2 |  | 5 | 2 |  | 7 | 1 |
| 6 | 1 |  | **5** | **1** |  | 6 | 1 |  | 7 | 6 |
| 7 | 6 |  | 5 | 2 |  | 7 | 6 |  | 8 | 1 |
| 8 | 6 |  | 6 | 1 |  | 8 | 6 |  | 8 | 6 |
| 9 | 6 |  | 7 | 1 |  | 9 | 6 |  | 9 | 1 |
|  |  |  | 7 | 6 |  | after del | residue |  | 9 | 6 |
|  |  |  | 8 | 1 |  |  | subquery | 1 | 2 | 7 |
|  |  |  | 8 | 6 |  |  |  |  | 2 | 1 |
|  |  |  | 9 | 1 |  |  | subquery | 2 | 2 | 6 |
|  |  |  | 9 | 6 |  |  |  |  | 3 | 7 |
|  |  |  |  |  |  |  |  |  | 4 | 7 |
|  |  |  |  |  |  |  | subquery | 3 | 5 | 7 |
|  |  |  |  |  |  |  |  |  | 3 | 1 |
|  |  |  |  |  |  |  |  |  | 3 | 6 |
|  |  |  |  |  |  |  |  |  | 4 | 1 |
|  |  |  |  |  |  |  | subquery | 4 | 4 | 6 |
|  |  |  |  |  |  |  |  |  | 5 | 1 |
|  |  |  |  |  |  |  |  |  | 5 | 6 |

**Phase 1**

To delete rows where Id belongs to the sub tree of *x*, and AncestorId belongs to the path of *x*

(*x* ∪ *subtree*(*x*))⨯ *path*(*x*)

*x* = 2

|  |  |
| --- | --- |
| **node** | **ancestor** |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |

*path*(*x*) = SELECT *Ancestor* FROM *t\_path* WHERE *Node* = *x*

*path*(2) = (1)

*subtree*(2) = SELECT *Id* FROM *t\_path* WHERE *AncestorId = x*

*subtree*(2) = (3, 4, 5)

(2∪ *subtree*(2)) = (2, 3, 4, 5)

Result of (2∪ *subtree*(2))⨯ *path*(2)

|  |  |
| --- | --- |
| **node** | **ancestor** |
| 2 | 7 |
| 3 | 7 |
| 4 | 7 |
| 5 | 7 |
| 2 | 6 |
| 3 | 6 |
| 4 | 6 |
| 5 | 6 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |

**Phase 2**

To insert *x* and the sub tree of *x*, together with *y* and the path of *y*

(*y* – node to attach to)

(*x* ∪ *subtree*(*x*))⨯(*y* ∪ *path*(*y*))

(*x* ∪ *subtree*(*x*)) = (2∪ *subtree*(2)) = (2, 3, 4, 5)

*y* = 7

*path*(7) = (6, 1)

(*y* ∪ *path*(*y*)) = (7, 6, 1)

Result of (*2* ∪ *subtree*(2))⨯(7∪ *path*(7))

**Phase 1 – SQL construction**

(*x* ∪ *subtree*(*x*))⨯ *path*(*x*)

DELETE FROM *t\_path* WHERE

((*node* = *x*) *x*

OR ∪

*node* IN (SELECT *node* FROM *t\_path* WHERE *ancestor* = *x*) *subtree*(*x*)

) AND ⨯

*ancestor* IN (SELECT *ancestor* FROM *t\_path* WHERE *node* = *x*) *path*(*x*)

**Phase 2 – SQL construction**

(*x* ∪ *subtree*(*x*))⨯(*y* ∪ *path*(*y*))

INSERT INTO *t\_path* (

(VALUES *x*, *y*) (*x*, *y*)

UNION

(SELECT x, *ancestor* FROM *t\_path* WHERE *node* = *y*) *x* ⨯ *path*(*y*)

UNION

(SELECT *node*, y FROM *t\_path* WHERE *ancestor* = *x*) *subtree*(*x*) ⨯ *y*

UNION

(SELECT *a.node*, *b.ancestor* FROM *t\_path* *a*, *t\_path* *b* WHERE

a.*ancestor* = *x* AND b.*node* = *y*) {*u* ∣ *ancestor*(*u*) = *x*} ⨯ {*v* ∣ *node*(*v*) = *y*}

)

|  |  |  |  |
| --- | --- | --- | --- |
| **part of query** |  | **node** | **ancestor** |
| (VALUES x, y) | (*x*, *y*) | 2 | 7 |
| (SELECT x, *ancestor* FROM *t\_path* WHERE *node* = y) | *x* ⨯ *path*(*y*) | 2 | 6 |
| 2 | 1 |
| (SELECT *node*, y FROM *t\_path* WHERE *ancestor* = x) | *subtree*(*x*) ⨯ *y* | 3 | 7 |
| 4 | 7 |
| 5 | 7 |
| (SELECT *a.node*, *b.ancestor* FROM *t\_path* *a*, *t\_path* *b* WHERE  a.*ancestor* = *x* AND b.*node* = *y*) | {*u* ∣ *ancestor*(*u*) = *x*} ⨯ {*v* ∣ *node*(*v*) = *y*} | 3 | 6 |
| 4 | 6 |
| 5 | 6 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |

Type combination and press Alt+X

222a = set union, ∪

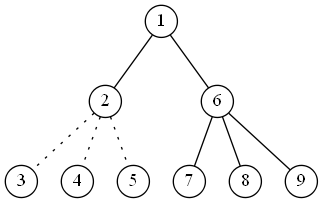
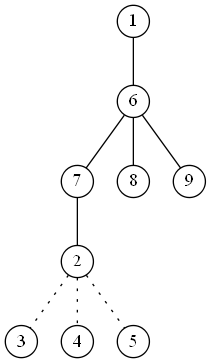
2a2f = dot product, ⨯

2297 = circled times, ⊗

2223 = such that, ∣

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall

Grid = page layout tab - arrange

**6. Move subtree**

Looks like we need parent-to-root order, due to residue from deletion

Source graph Resulting graph

t\_data t\_path t\_data t\_path

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **node** | **parent** |  | **node** | **ancestor** |  | **node** | **ancestor** |  | **node** | **ancestor** |
| 1 | 1 |  | **2** | **1** |  | 1 | 1 |  | 3 | 2 |
| 2 | 1 |  | 3 | 2 |  | **2** | **7** |  | 4 | 2 |
| 3 | 2 |  | **3** | **1** |  | 3 | 2 |  | 5 | 2 |
| 4 | 2 |  | 4 | 2 |  | 4 | 2 |  | 6 | 1 |
| 5 | 2 |  | **4** | **1** |  | 5 | 2 |  | 7 | 6 |
| 6 | 1 |  | 5 | 2 |  | 6 | 1 |  | 7 | 1 |
| 7 | 6 |  | **5** | **1** |  | 7 | 6 |  | 8 | 6 |
| 8 | 6 |  | 6 | 1 |  | 8 | 6 |  | 8 | 1 |
| 9 | 6 |  | 7 | 6 |  | 9 | 6 |  | 9 | 6 |
|  |  |  | 7 | 1 |  | after del | residue |  | 9 | 1 |
|  |  |  | 8 | 6 |  |  | subquery | 1 | 2 | 7 |
|  |  |  | 8 | 1 |  |  |  |  | 2 | 6 |
|  |  |  | 9 | 6 |  |  | subquery | 2 | 2 | 1 |
|  |  |  | 9 | 1 |  |  |  |  | 3 | 7 |
|  |  |  |  |  |  |  |  |  | 4 | 7 |
|  |  |  |  |  |  |  | subquery | 3 | 5 | 7 |
|  |  |  |  |  |  |  |  |  | 3 | 6 |
|  |  |  |  |  |  |  |  |  | 3 | 1 |
|  |  |  |  |  |  |  |  |  | 4 | 6 |
|  |  |  |  |  |  |  | subquery | 4 | 4 | 1 |
|  |  |  |  |  |  |  |  |  | 5 | 6 |
|  |  |  |  |  |  |  |  |  | 5 | 1 |

**Phase 1**