

Data Model: Relationships



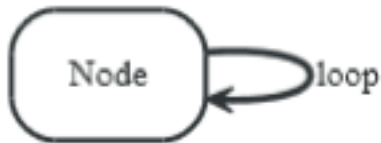
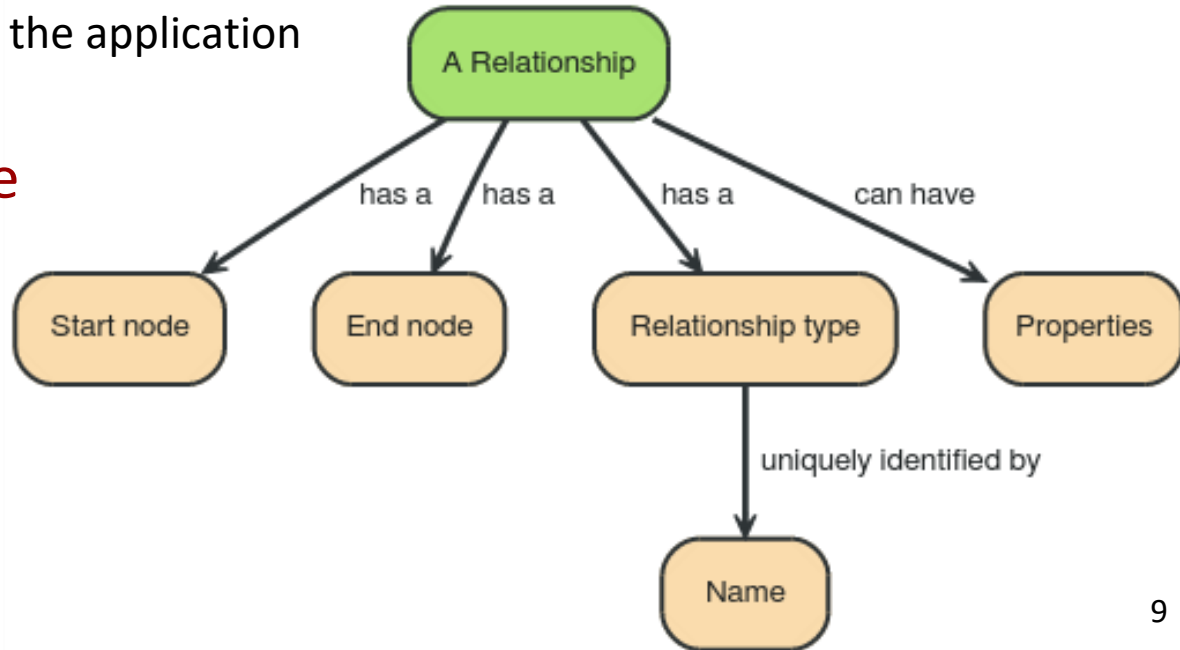
- Directed relationships (edges)

- Incoming and outgoing **edge**

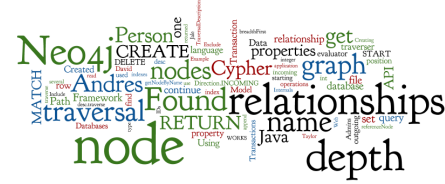
- Equally **efficient traversal** in both directions
 - Direction **can be ignored** if not needed by the application

- Always **a start** and **an end node**

- Can be recursive



Neo4j Command-line Querying



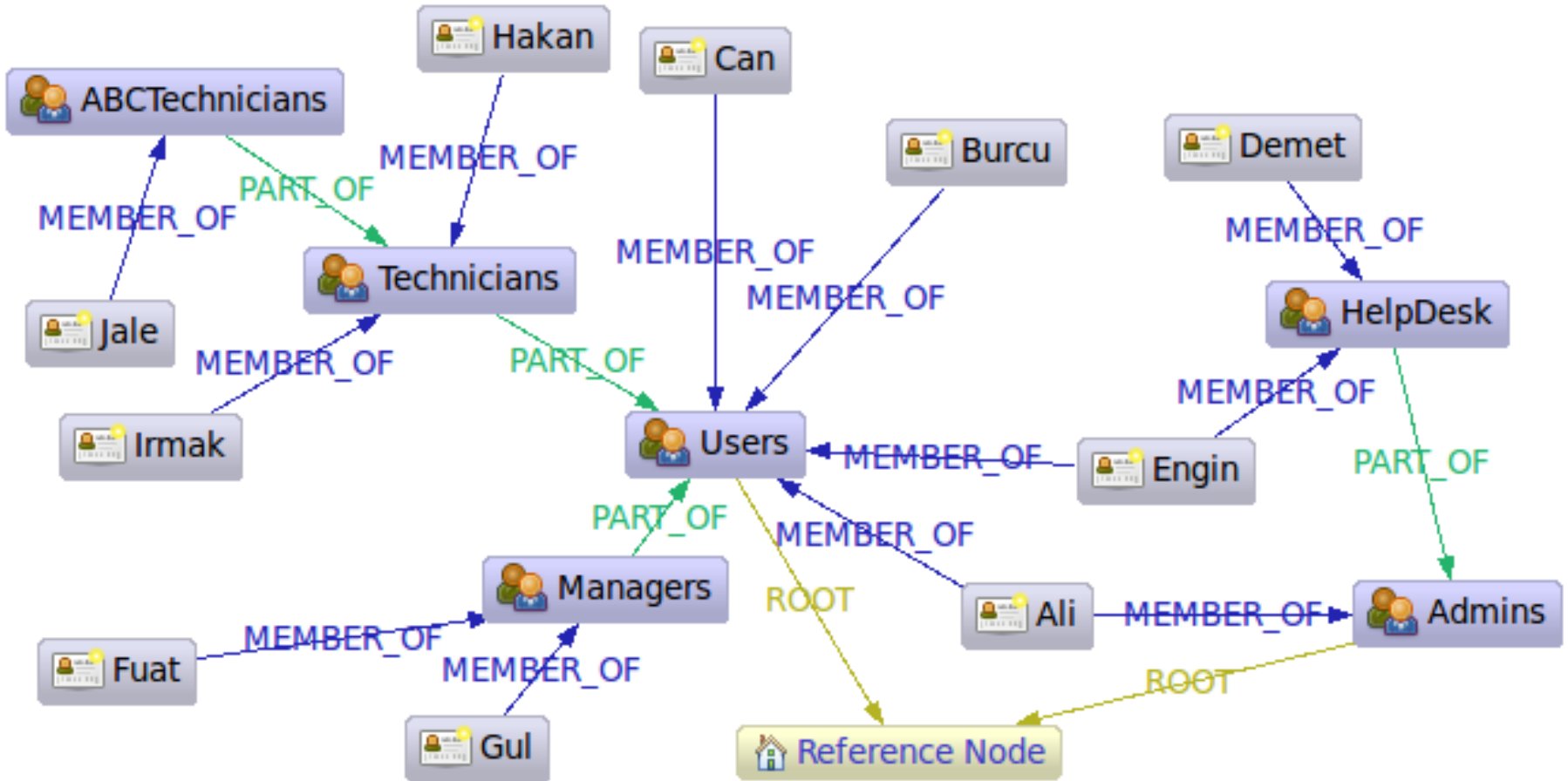
- **Cypher shell**
 - `./bin/cypher-shell`
 - can also be installed separately, but shipped with the server

Traversal Framework – Java API



- `org.neo4j...TraversalDescription`
 - The main **interface** for defining **traversals**
 - Can specify branch ordering `breadthFirst()` / `depthFirst()`
- `.relationships()`
 - Specify the **relationship types** to traverse
 - e.g., traverse only edge types: `FRIEND`, `RELATIVE`
 - Empty (default) = traverse all relationships
 - Can also specify **direction**
 - `Direction.BOTH`
 - `Direction.INCOMING`
 - `Direction.OUTGOING`

Sample Data



Query: Get Group Membership of a User

```
Node jale = getNodeByName( "Jale" );
desc = graphDb.traversalDescription()
    .depthFirst()
    .evaluator( Evaluators.excludeStartPosition() )
    .relationships( RoleRels.MEMBER_OF, Direction.OUTGOING )
    .relationships( RoleRels.PART_OF, Direction.OUTGOING );

traverser = traversalDescription.traverse( jale );
```

```
Found: ABCTechnicians at depth: 1
Found: Technicians at depth: 2
Found: Users at depth: 3
```

Query: Get All Groups

```
Node referenceNode = getNodeByName( "Reference_Node" ) ;
desc = graphDb.traversalDescription()
    .breadthFirst()
    .evaluator( Evaluators.excludeStartPosition() )
    .relationships(RoleRels.ROOT, Direction.INCOMING )
    .relationships(RoleRels.PART_OF, Direction.INCOMING);

traverser = desc.traverse( referenceNode );
```

```
Found: Admins at depth: 1
Found: Users at depth: 1
Found: HelpDesk at depth: 2
Found: Managers at depth: 2
Found: Technicians at depth: 2
Found: ABCTechnicians at depth: 3
```

Query: Get All Members in the Database

```
Node referenceNode = getNodeByName( "Reference_Node" ) ;
desc = graphDb.traversalDescription()
    .breadthFirst()
    .evaluator(Evaluators.includeWhereLastRelationshipTypeIs
        (RoleRels.MEMBER_OF ) );

traverser =
desc.traverse( referenceNode );
```

```
Found: Ali at depth: 2
Found: Engin at depth: 2
Found: Burcu at depth: 2
Found: Can at depth: 2
Found: Demet at depth: 3
Found: Gul at depth: 3
Found: Fuat at depth: 3
Found: Hakan at depth: 3
Found: Irmak at depth: 3
Found: Jale at depth: 4
```