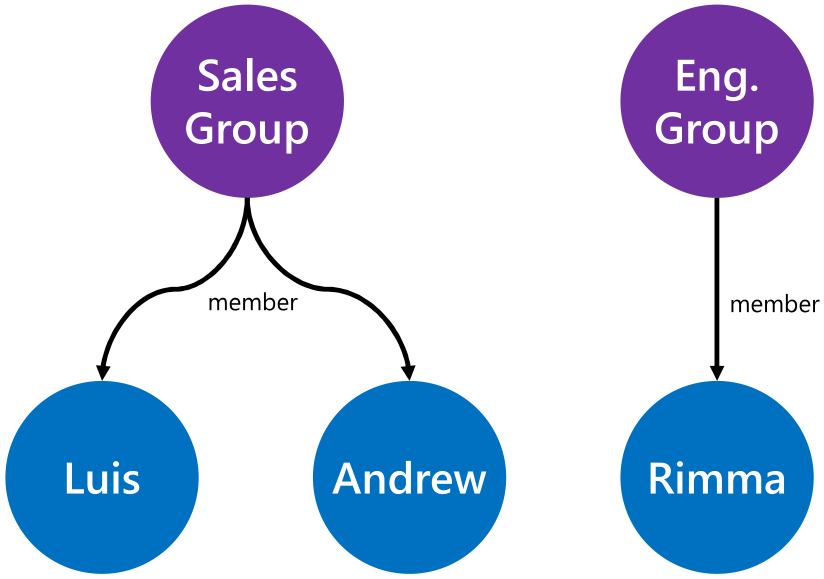
Chapter 1: The Green-field app

Human Resources Data

Relational vs Graph oriented model comparison

Employee ID	Name	Group
1	Luis Bosquez	Sales
2	Rimma Nehme	Engineering
3	Andrew Liu	Sales



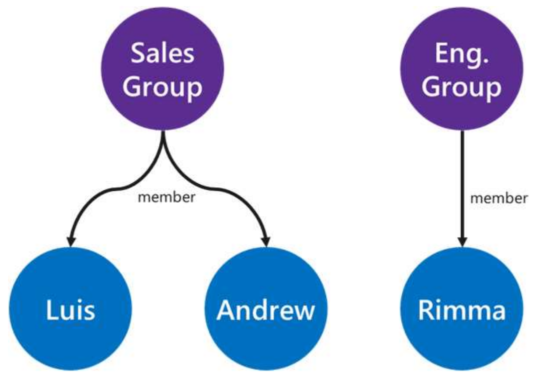
3 rows, 3 columns

8 documents (vertices and edges)

Human Resources Data

Query comparison: get all employees.

Employee ID	Name	Group
1	Luis Bosquez	Sales
2	Rimma Nehme	Engineering
3	Andrew Liu	Sales



```
SELECT * FROM v1_Employees;
```

```
g.V().hasLabel('employee')
```

BREAKING NEWS
That company you work for will have a major reorg
World Bank's stock at all-time high
oh no

Chapter 2: The hidden business requirement

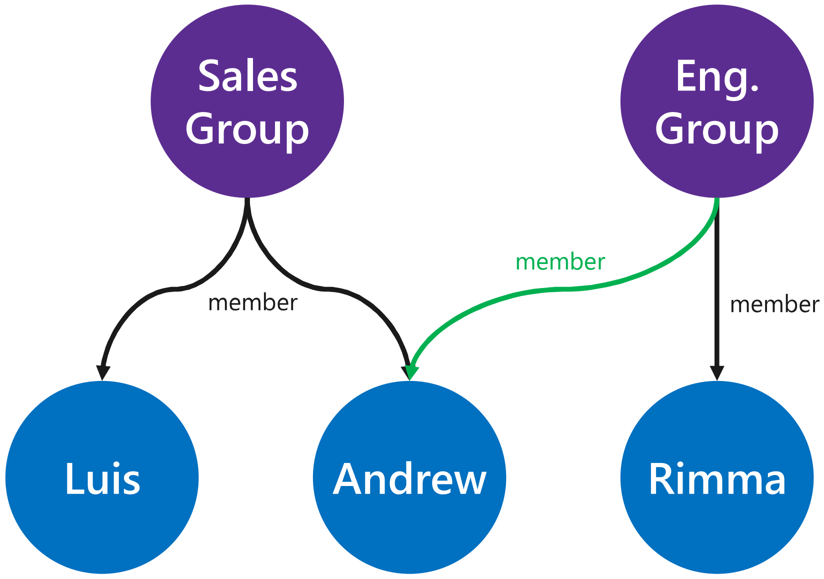
REORG TIME

Employees can now belong to multiple groups

Employee ID	Name
1	Luis B.
2	Rimma N.
3	Andrew L.

Group ID	Name
1	Sales
2	Engineering

FK Employee ID	FK Group ID
1	1
2	2
3	1
3	2



+ 2 tables, 6 rows, 4 new columns,
-1 column alteration

+ 1 document

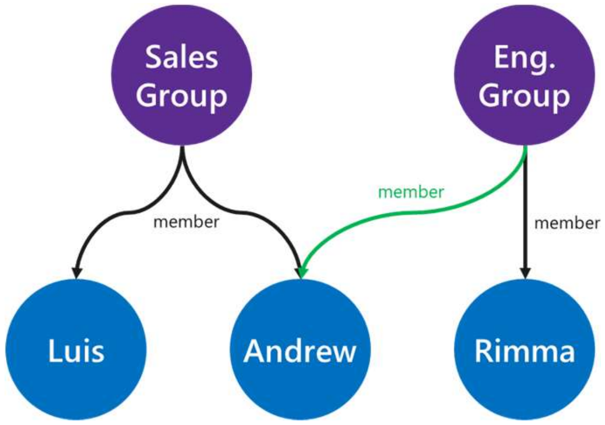
REORG TIME

Query comparison: get employees from the Sales group.

Employee ID	Name
1	Luis B.
2	Rimma N.
3	Andrew L.

Group ID	Name
1	Sales
2	Engineering

FK Employee ID	FK Group ID
1	1
2	2
3	1
3	2



```
SELECT * FROM v2_Employees
INNER JOIN v2_Employee_Group eg
    ON Employee_ID = FK_Employee_ID
INNER JOIN v2_Groups g
    ON FK_Group_ID = Group_ID
WHERE g.Group_Name = 'Sales'
```

```
g.V('sales')
       .out('member')
```

pls, no
Your organization has merged with another one

Chapter 3: The unexpected data migration

Nested groups

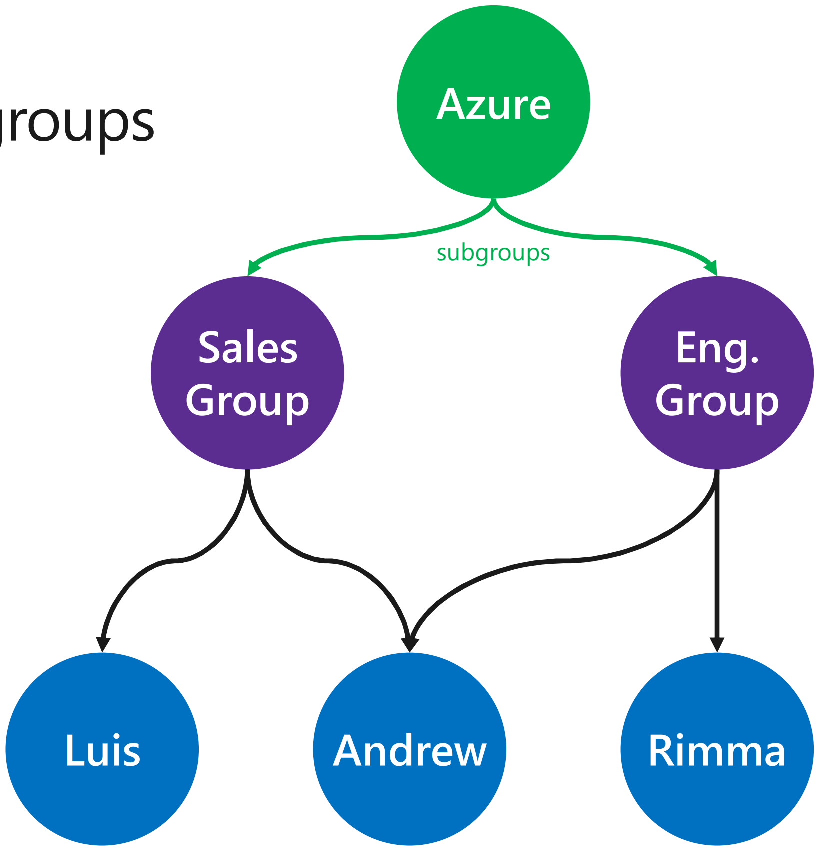
Employees are now part of multi-level groups

Employee ID	Name
1	Luis B.
2	Rimma N.
3	Andrew L.

Group ID	Name
1	Sales
2	Engineering
3	Azure

FK Group ID	FK Nested Group ID
1	3
2	3

FK Employee ID	FK Group ID
1	1
2	2
3	1
3	2
1	3
1	3
1	3



+ 1 tables, 6 rows, 2 new columns

+ 3 documents

Nested groups

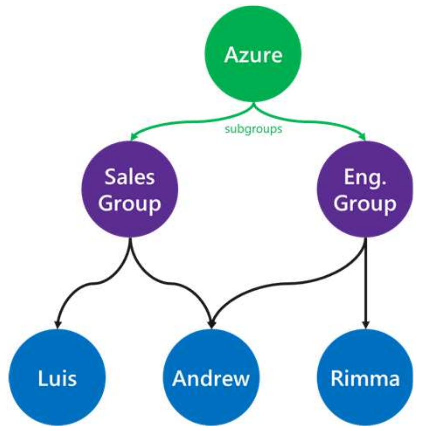
Query comparison: get all groups under the Azure group

Employee ID	Name
1	Luis B.
2	Rimma N.
3	Andrew L.

Group ID	Name
1	Sales
2	Engineering
3	Azure

FK Group ID	FK Nested Group ID
1	3
2	3

FK Employee ID	FK Group ID
1	1
2	2
3	1
3	2
1	3
1	3
1	3



```sql
SELECT g.Group_ID, g.Group_Name FROM v3_Groups g
INNER JOIN v3_Group_Group gg
ON gg.FK_Child_Group_ID = Group_ID
WHERE FK_Parent_Group_ID =
(SELECT Group_ID FROM v3_Groups WHERE
Group_Name='Azure')
```

```
g.V('Azure')
      .out('subgroup')
```

Imagine a revenue of hundred million dollars
The world is messed up
but I believe we are in a position now to help
Fact about online marketing
Million reinvested in travel businesses

You, yes you, will now have an additional manager. Effective immediately.

haha

why is this on the news?

Chapter 4: The overloaded workload

Additional hierarchies

More than one hierarchical structure is represented in the database

Employee ID	Name
1	Luis B.
2	Rimma N.
3	Andrew L.

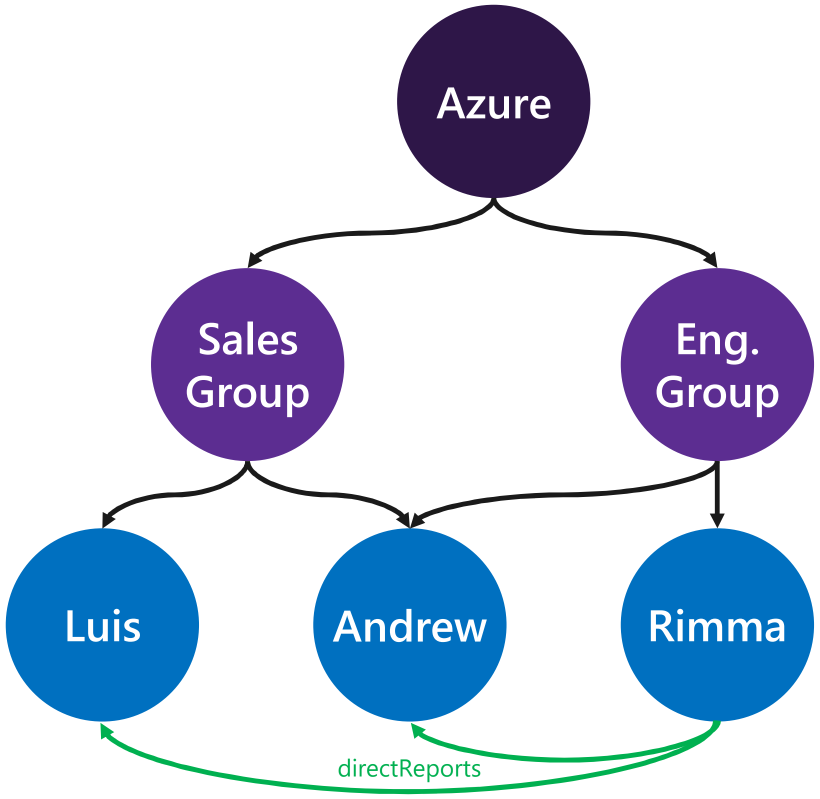
FK Employee ID	FK Group ID
1	1
2	2
3	1
3	2
1	3
1	3
1	3

Group ID	Name
1	Sales
2	Engineering
3	Azure

FK Group ID	FK Nested Group ID
1	3
2	3

FK Employee ID	FK Report Employee ID
2	1
2	2

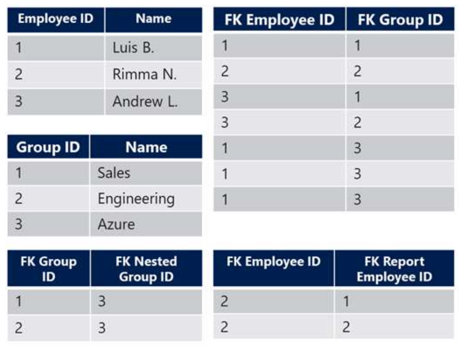
+ 1 table, 2 rows, 2 new columns



+ 2 documents

Additional hierarchies

Query comparison: Obtain all managers from the Engineering Group



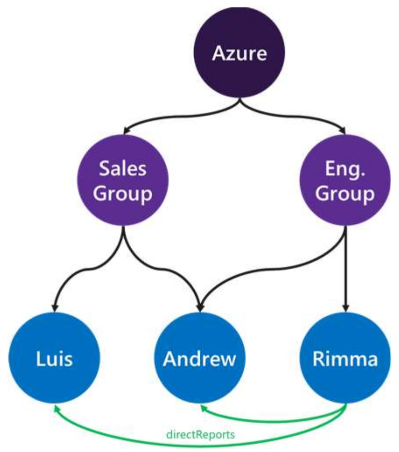
Employee ID	Name
1	Luis B.
2	Rimma N.
3	Andrew L.

FK Employee ID	FK Group ID
1	1
2	2
3	1
3	2
1	3
1	3
1	3

Group ID	Name
1	Sales
2	Engineering
3	Azure

FK Group ID	FK Nested Group ID
1	3
2	3

FK Employee ID	FK Report Employee ID
2	1
2	2



```sql
SELECT DISTINCT Employee_Name FROM v4_Employees e
INNER JOIN v4_Employee_Group eg
    ON eg.FK_Employee_ID = e.Employee_ID
INNER JOIN v4_Employee_Employee ee
    ON ee.FK_Parent_Employee_ID = e.Employee_ID
WHERE eg.FK_Group_ID = (
    SELECT g.Group_ID FROM v4_Groups g
    WHERE g.Group_Name = 'Engineering'
)
```

```
g.V('engineering')
        .out('members')
        .in('has_report')
        .values('id')
```