Dungeons, Dragons, and Graph Databases

redis

Guy Royse

Developer Advocate

redis

@guyroyse

github.com/guyroyse

guy.dev

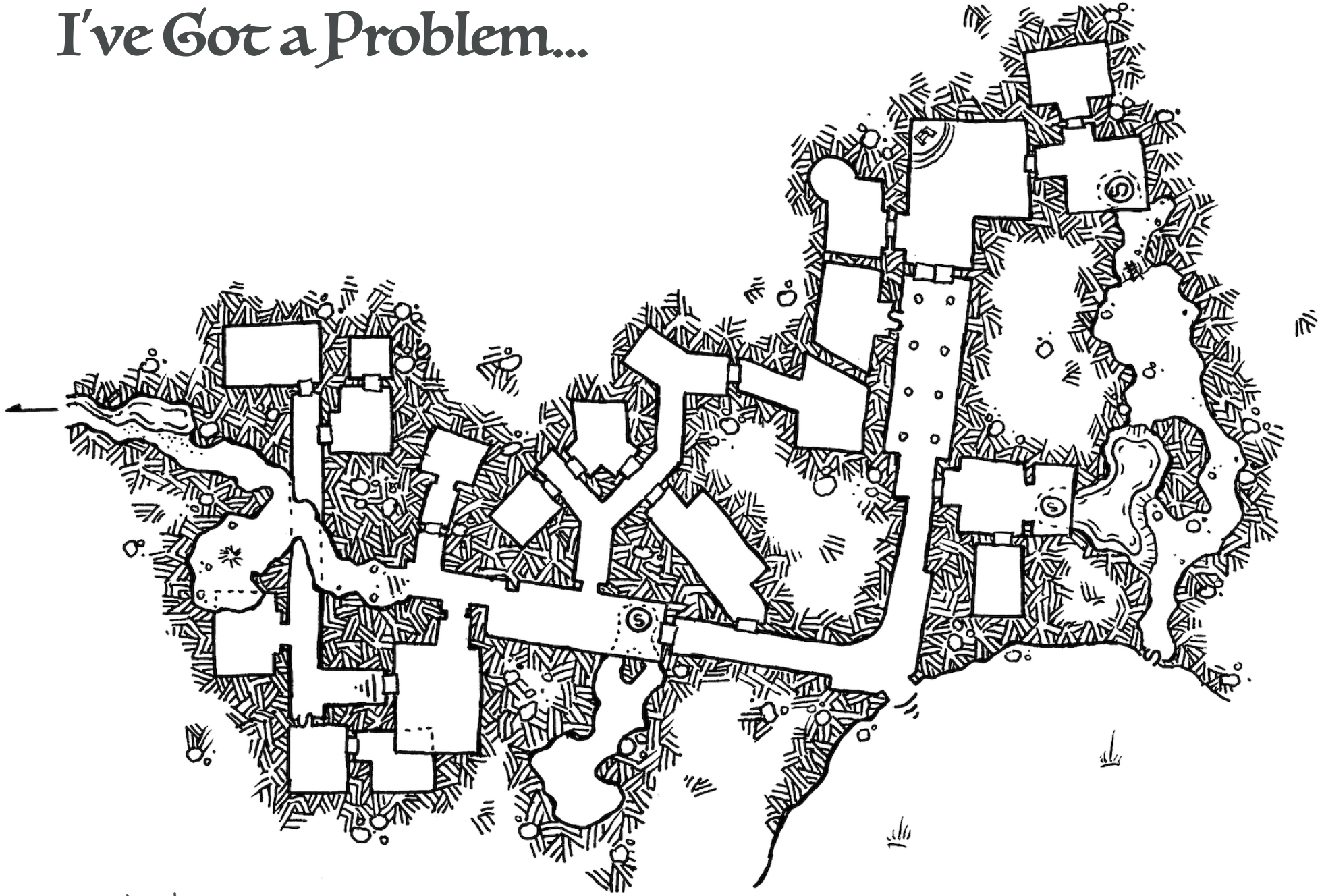
Name: Guy Royse
Alignment: Neutral good

Class: Bard
Race: Dwarf
Level: 5

AC: 11
HP: 32

Str: 10
Dex: 8
Con: 12
Int: 8
Wis: 8
Cha: 16

I've Got a Problem...



Cartography by Dyson Logos

redis

...using a Relational Database

Rooms

id: int
name: string

Treasures

id: int
name: string
gp: int
room_id: int

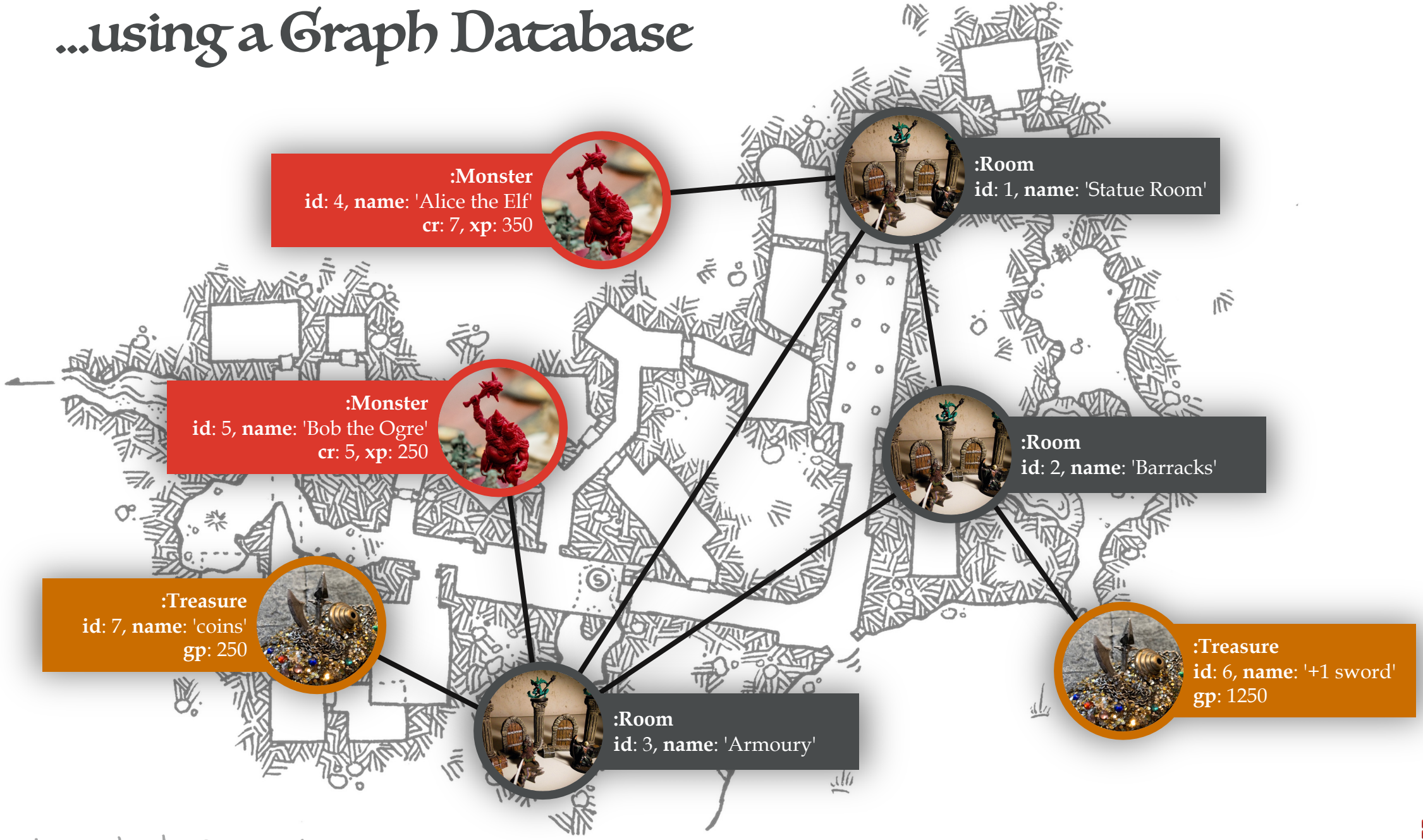
Monsters

id: int
name: string
cr: int
xp: int
room_id: int

Cartography by Dyson Logos

redis

...using a Graph Database



Cartography by Dyson Logos

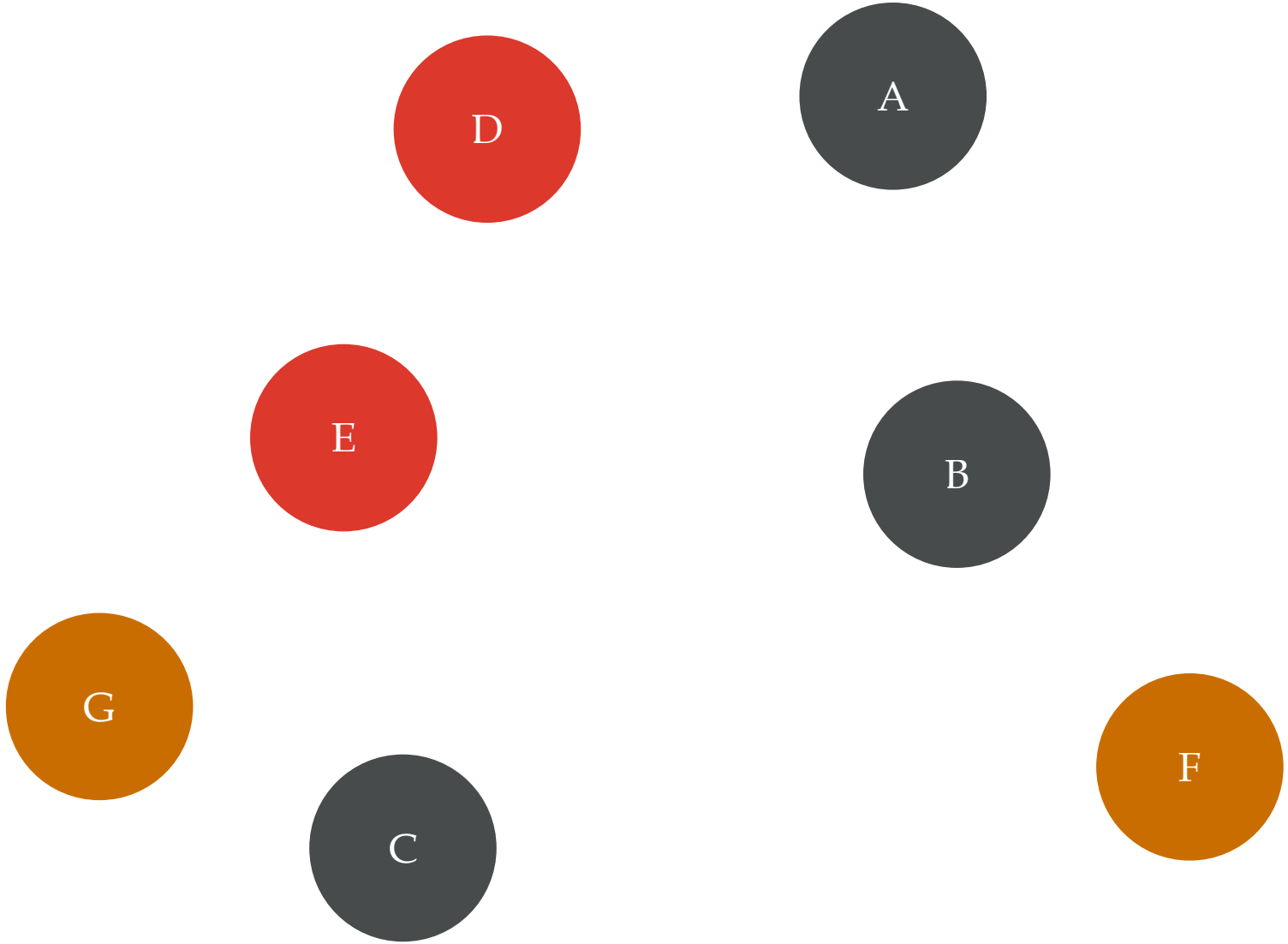
redis

What's a Graph Database?

What's a Graph?

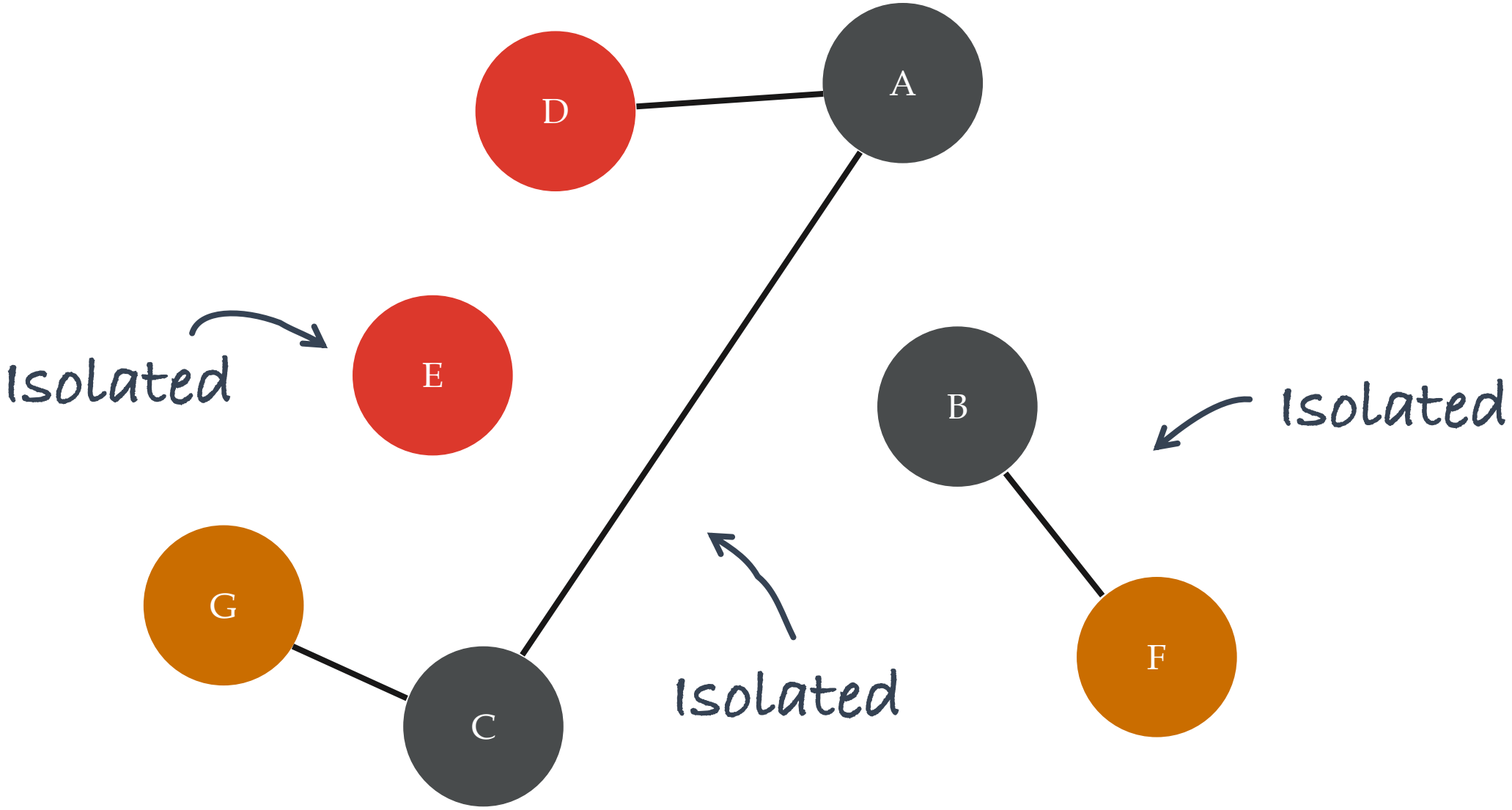
Null Graph

Add Some Nodes



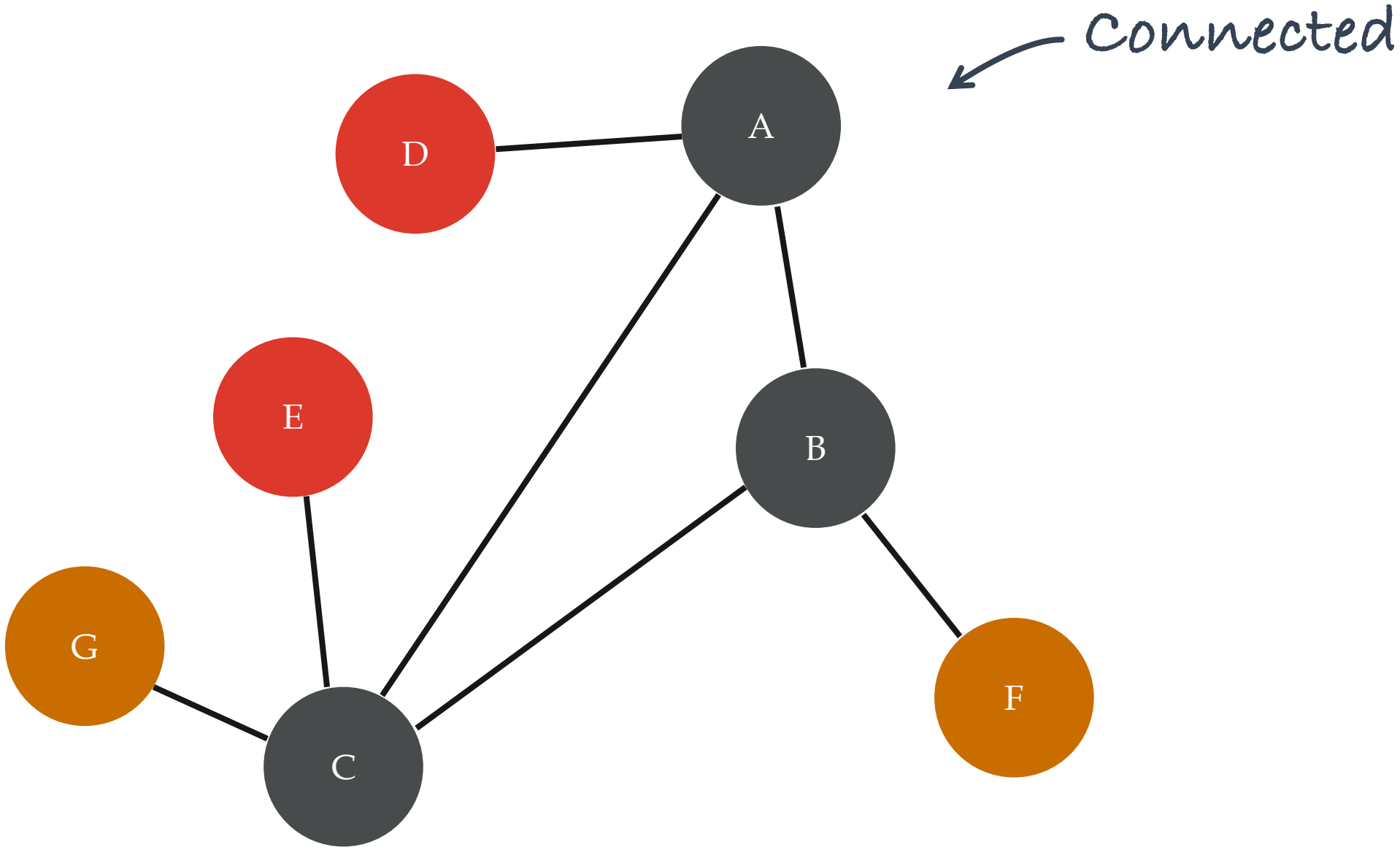
redis

Add Some Relationships



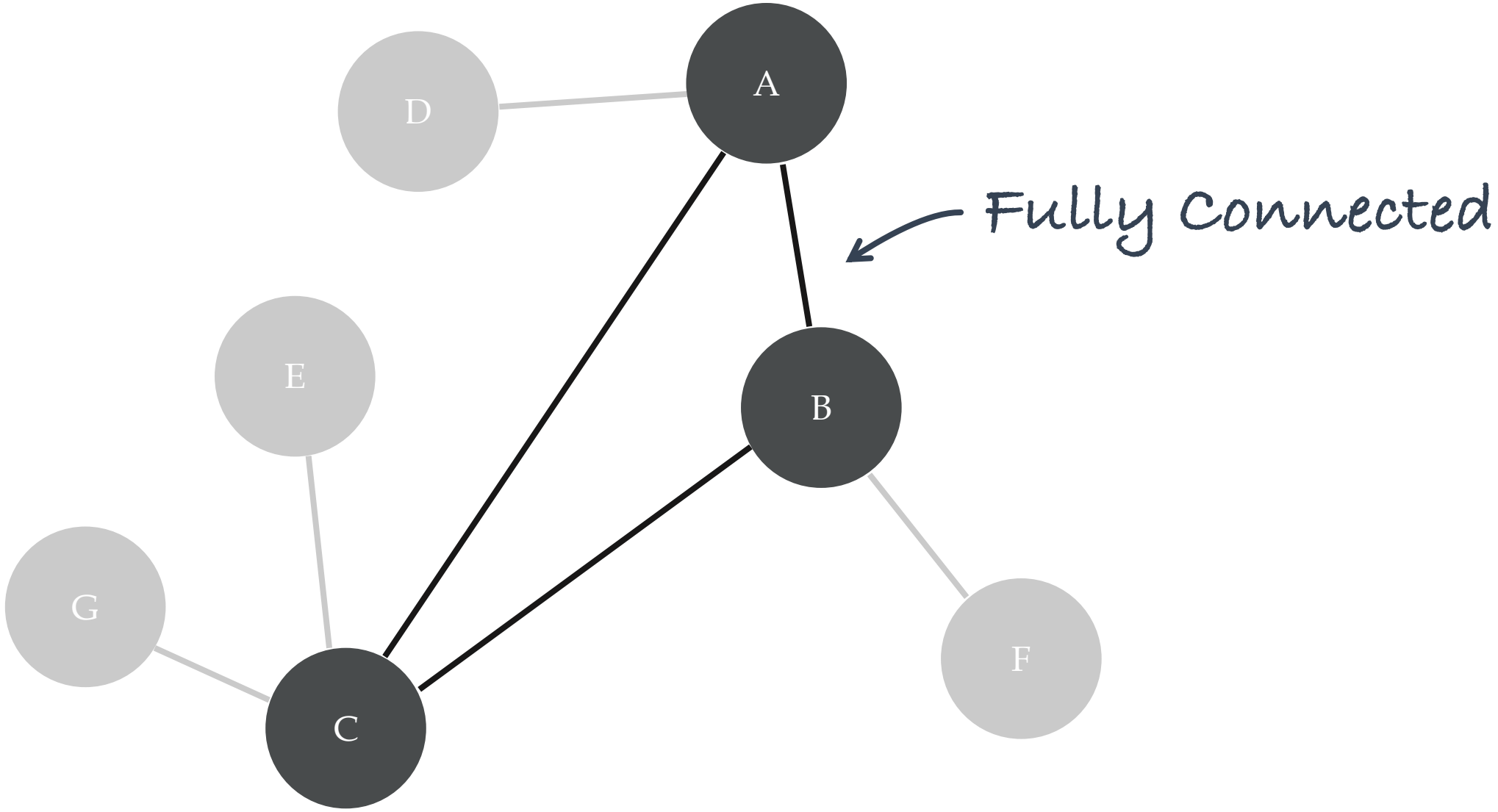
redis

Connect All the Nodes



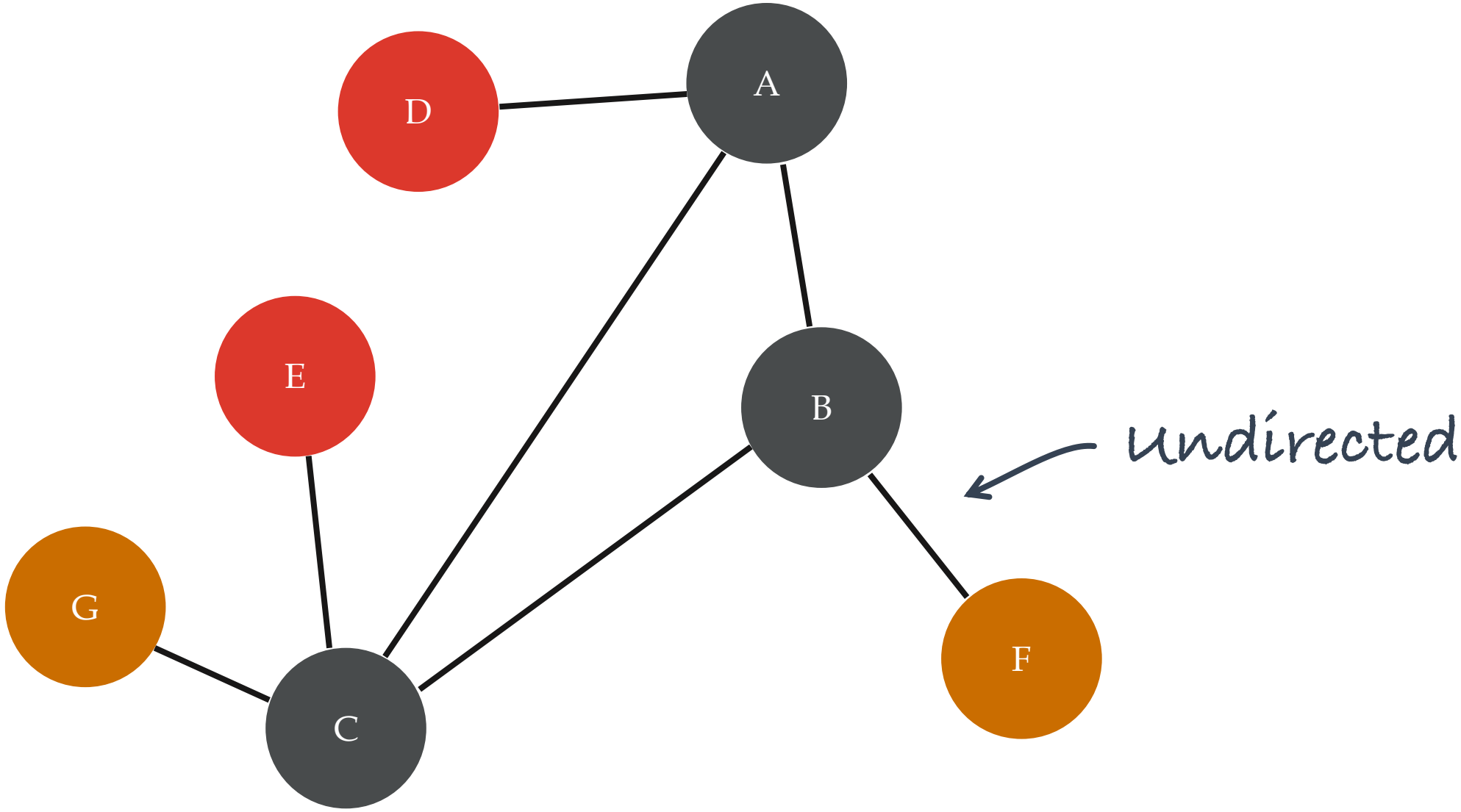
redis

Fully Connected



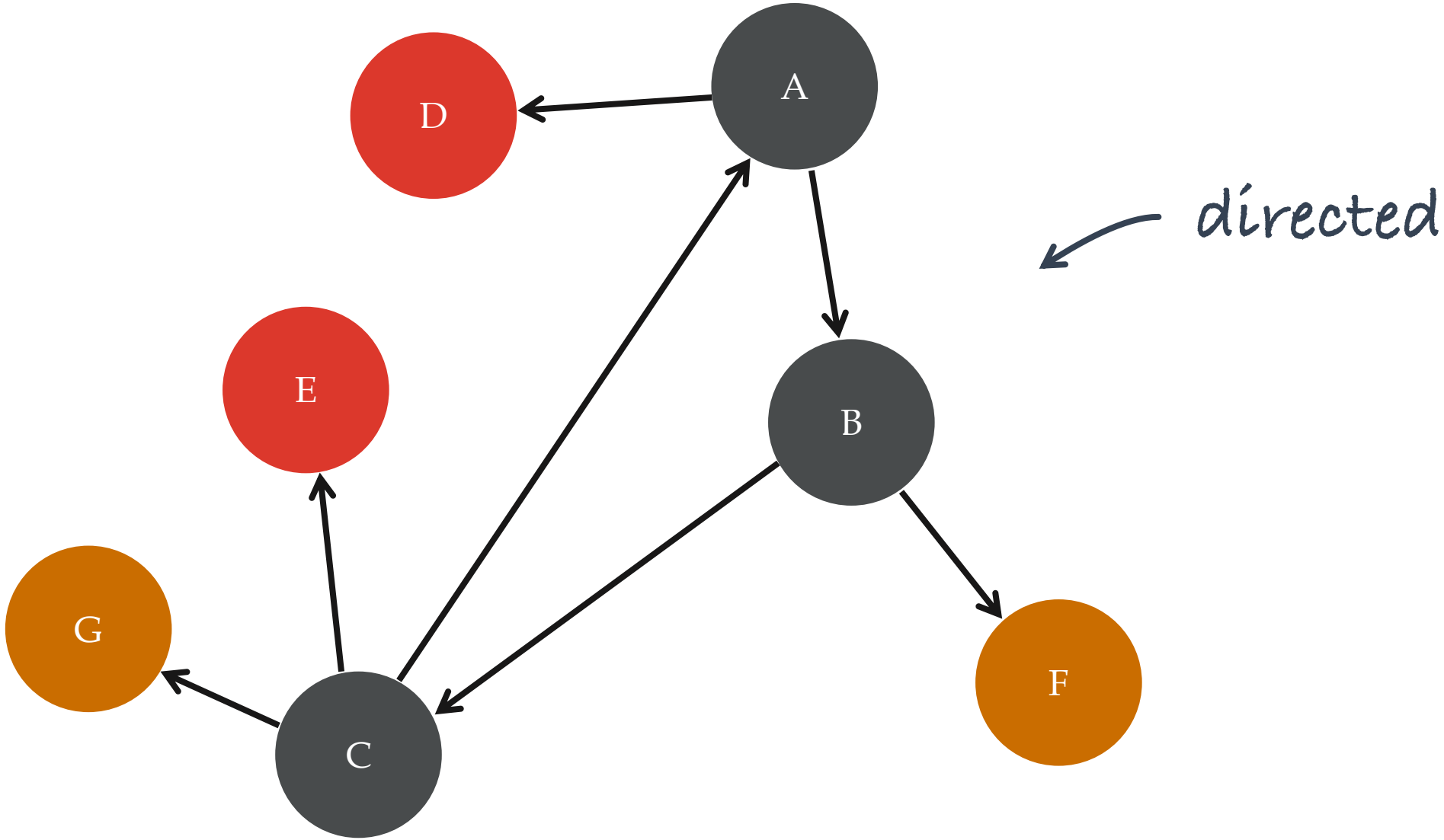
redis

An Undirected Graph



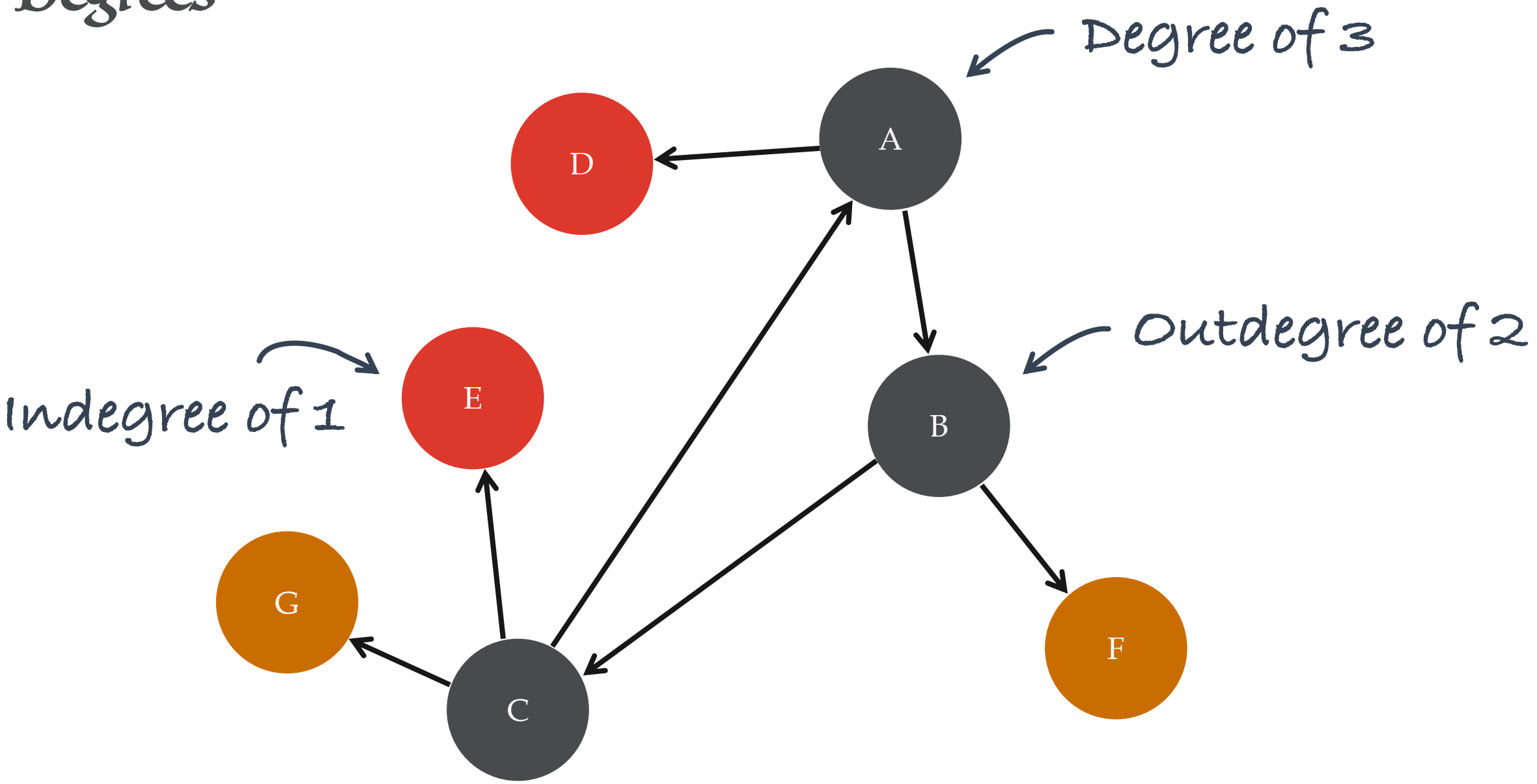
redis

A Directed Graph



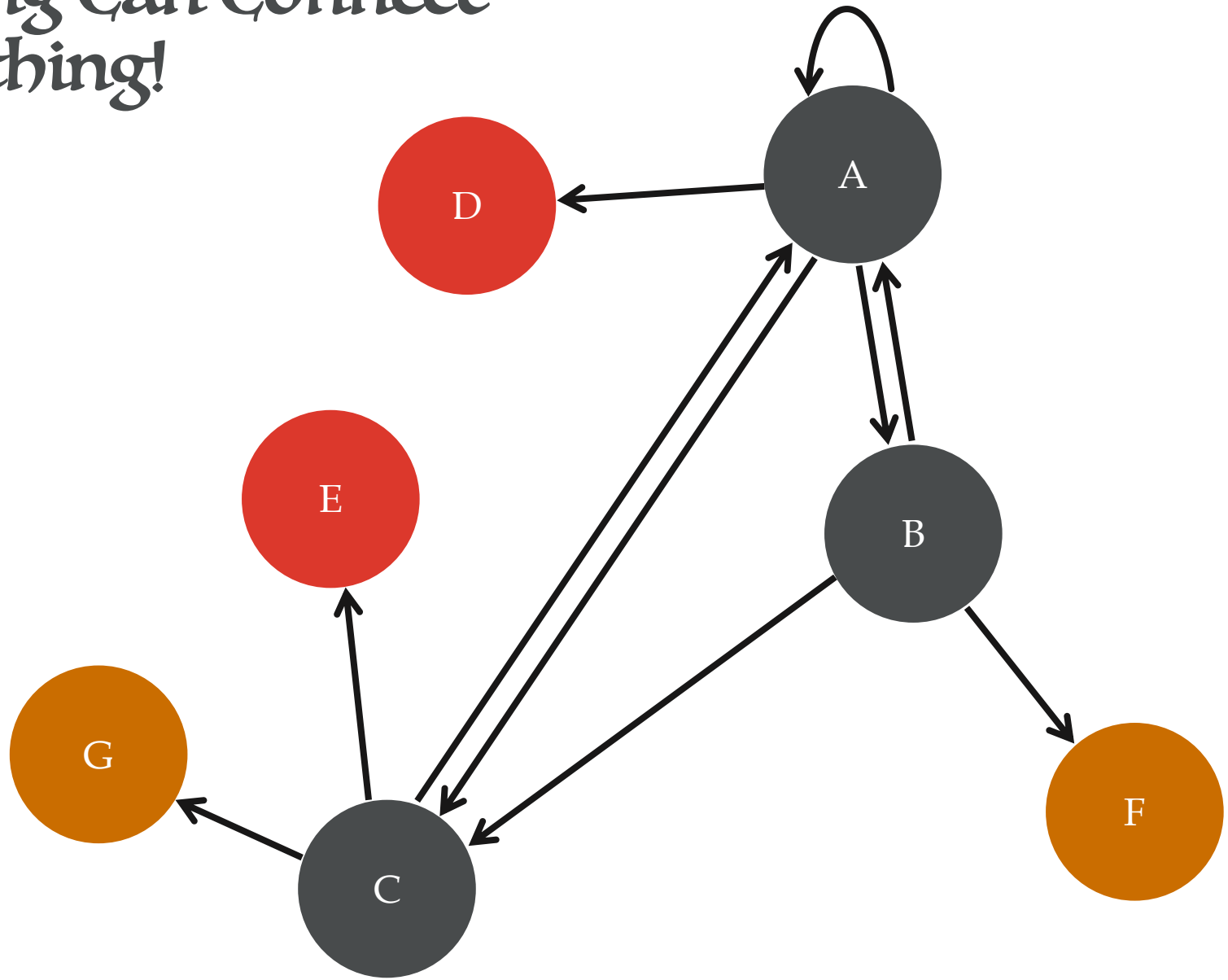
redis

Degrees



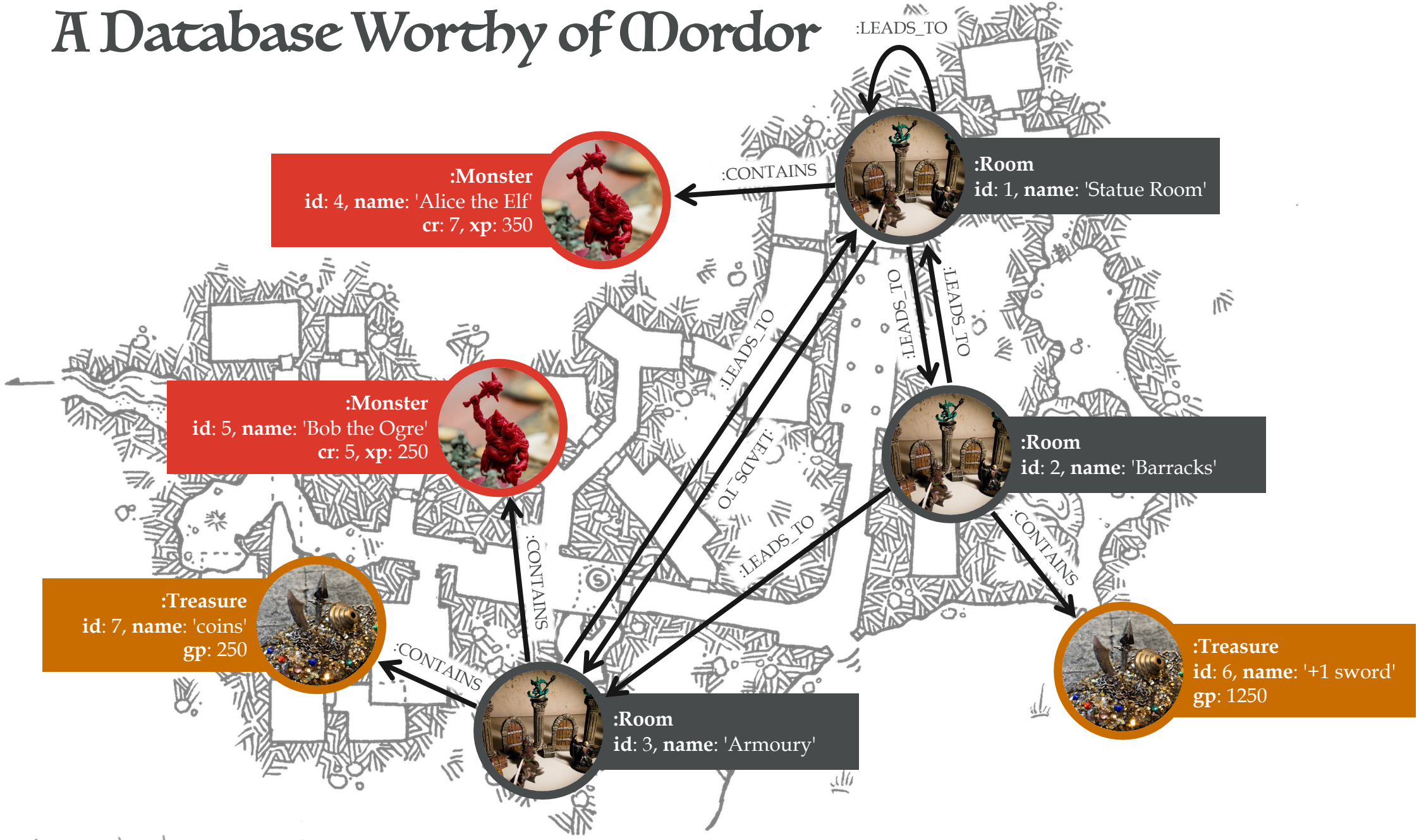
redis

Anything Can Connect to Anything!



redis

A Database Worthy of Mordor



Cartography by Dyson Logos

redis

Nodes & Relationships

Nodes

- Represent items
  - Can have labels
  - Can have attributes
- Can stand alone

Relationships

- Represent connections
  - Connect two nodes
  - Have a type
  - Have a direction
  - Can have attributes
- Cannot exist without nodes to connect

redis

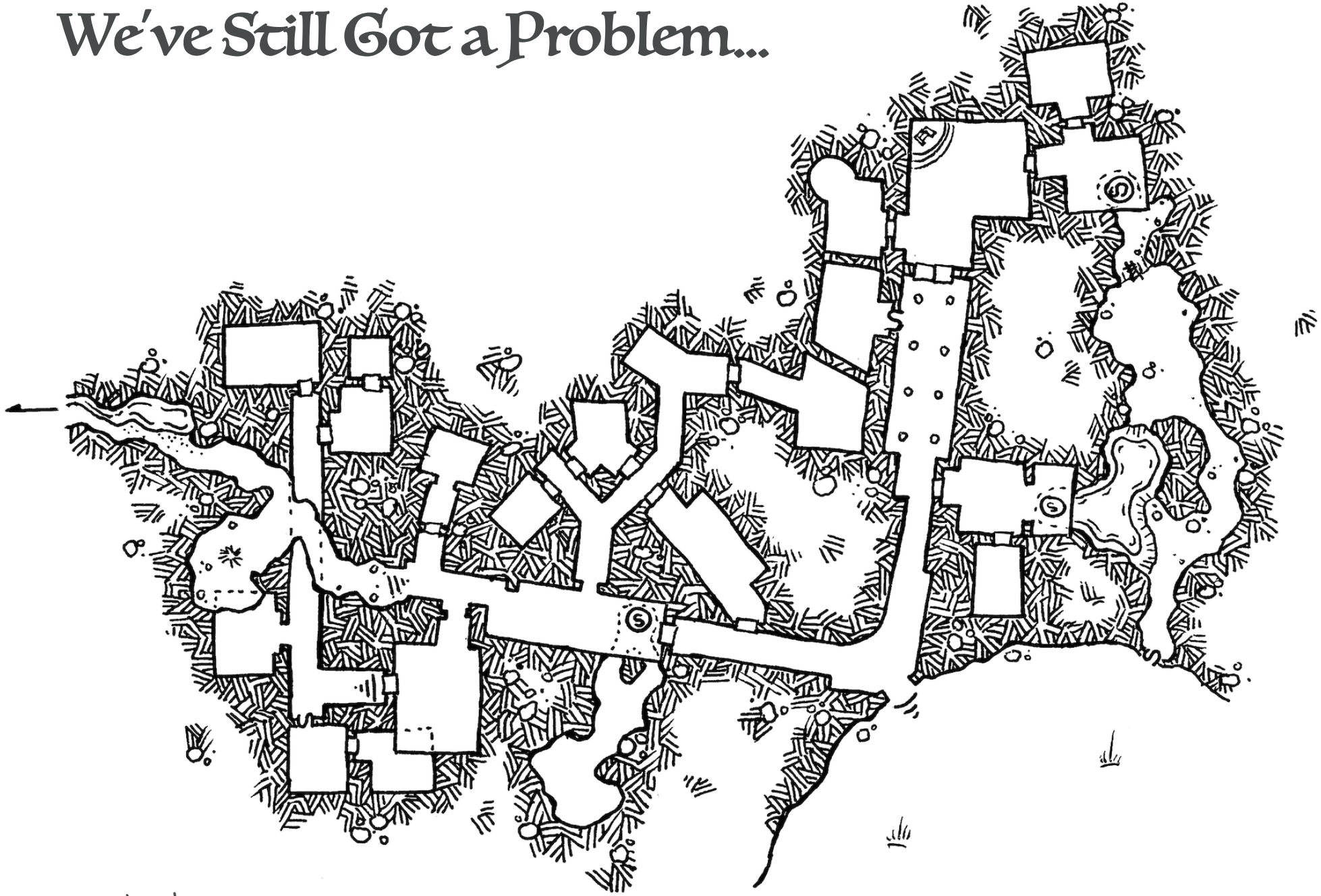
Reads Like a Sentence

The room contains a monster.

redis


Enough Theory!

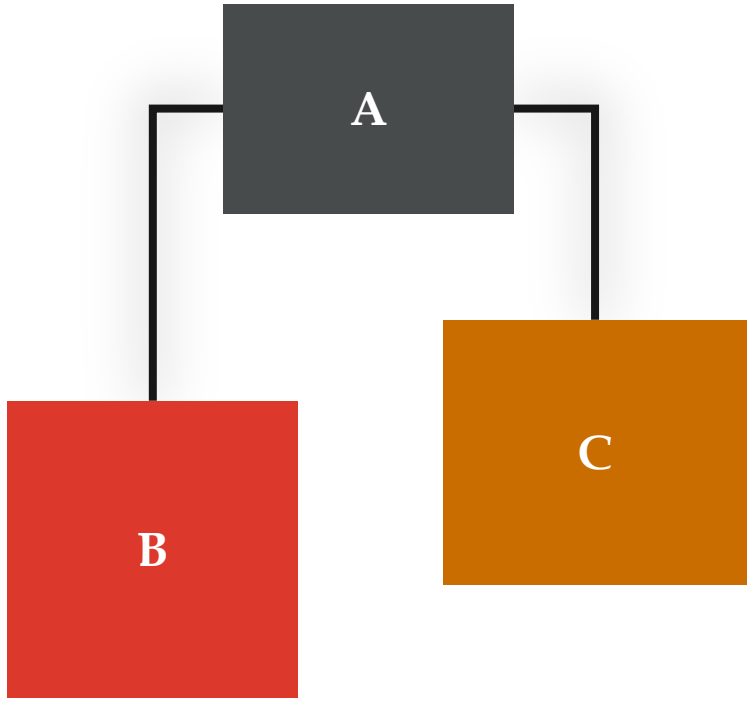
We've Still Got a Problem...

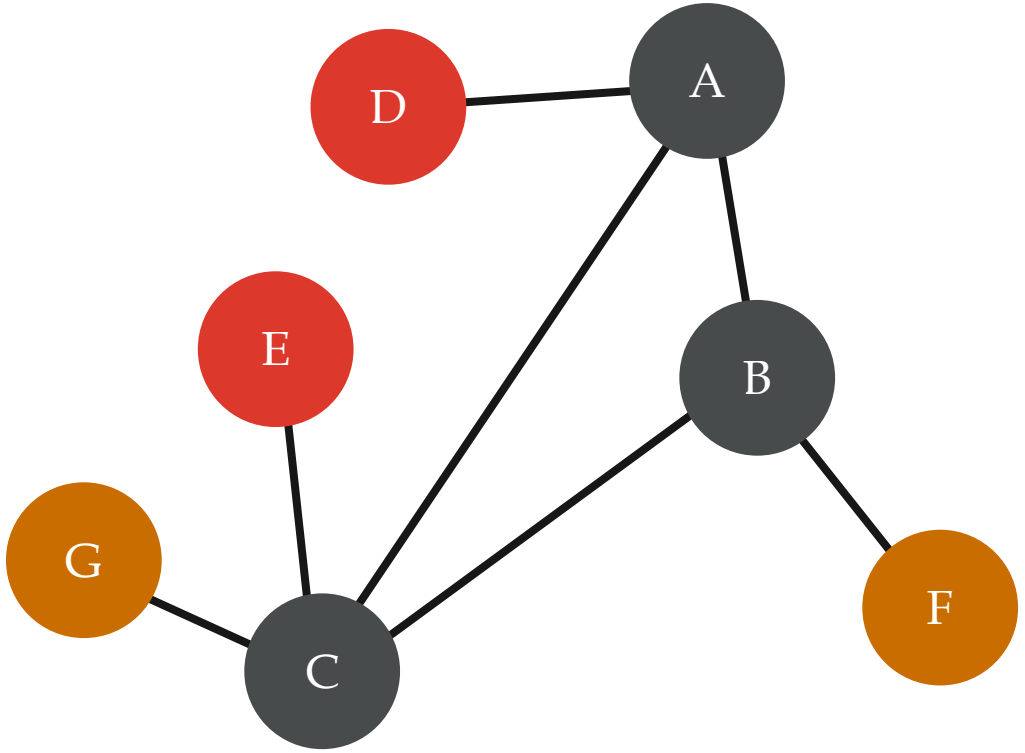


Cartography by Dyson Logos

redis

How We Gonna Solve It?





redis

Table Talk

SQL

```
SELECT
  r.id, r.name, t.id, t.name, t.gp m.id, m.name, m.cr, m.xp
FROM
  Rooms as r, Treasures as t, Monsters as m
WHERE
  t.room_id = r.id AND m.room_id = r.id;
```

Cypher

```
MATCH
  (t:Treasure)<-[:CONTAINS]-(r:Room)-[:CONTAINS]->(m:Monster)
RETURN
  r, t, m
```

Matching


```
(r:Room)-[c:CONTAINS]->(m:Monster)
```

matches room node

matches contains relationship

matches monster node

redis

Creating Rooms

Rooms	
id	name
1	Statue Room
2	Barracks
3	Armoury

:Room
id: 1, **name**: 'Statue Room'

:Room
id: 2, **name**: 'Barracks'

:Room
id: 3, **name**: 'Armoury'

```
INSERT INTO Rooms
  (id, name)
VALUES
  (1, 'Statue Room');
```

```
CREATE (r:Room)
  SET
    r.id = 1,
    r.name = 'Statue Room'
```

redis

Reading Rooms

Rooms	
id	name
1	Statue Room
2	Barracks
3	Armoury

:Room
id: 1, **name**: 'Statue Room'

:Room
id: 2, **name**: 'Barracks'

:Room
id: 3, **name**: 'Armoury'

```
SELECT id, name
FROM Rooms
WHERE id = 1;
```

```
MATCH (r:Room)
WHERE r.id = 1
RETURN r
```

redis

Updating Rooms

Rooms	
id	name
1	Statue Room
2	Barracks
3	Armoury

:Room
id: 1, **name**: 'Statue Room'

:Room
id: 2, **name**: 'Barracks'

:Room
id: 3, **name**: 'Armoury'

```
UPDATE Rooms
SET name = 'Statue Hall'
WHERE id = 1;
```

```
MATCH (r:Room)
WHERE r.id = 1
SET r.name = 'Statue Hall'
```

redis

Deleting Rooms

Rooms	
id	name
1	Statue Room
2	Barracks
3	Armoury

:Room
id: 1, **name**: 'Statue Room'

:Room
id: 2, **name**: 'Barracks'

:Room
id: 3, **name**: 'Armoury'

```
DELETE FROM Rooms
WHERE id = 1;
```

```
MATCH (r:Room)
WHERE r.id = 1
DELETE r
```

redis

A Note on Nodes

:Room
id: 1, **name**: 'Statue Room'

:Room
id: 2, **name**: 'Barracks'

:Room
id: 3, **name**: 'Armoury'

```
MATCH (r:Room) WHERE r.id = 1
```

```
MATCH (r:Room { id: 1 })
```

```
CREATE (r:Room)
SET r.id = 1, r.name = 'Statue Room'
```

```
CREATE (:Room {
  id: 1, name: 'Statue Room'
})
```

redis

Creating Monsters

Monsters				
	id	name	cr	xp
	4	Alice the Elf	7	350
	5	Bob the Ogre	5	250

:Monster
id: 4, **name**: 'Alice the Elf'
cr: 7, **xp**: 350

:Monster
id: 5, **name**: 'Bob the Ogre'
cr: 5, **xp**: 250

```
INSERT INTO Monsters
  (id, cr, xp, name)
VALUES (4, 7, 350,
  'Alice the Elf');
```

```
CREATE (:Monster {
  id: 4, cr: 7, xp: 350,
  name: 'Alice the Elf' })
```

redis

Creating Treasure

Treasures		
id	name	gp
6	+1 sword	1250
7	coins	250

:Treasure
id: 6, **name**: '+1 sword'
gp: 1250

:Treasure
id: 7, **name**: 'coins'
gp: 250

```
INSERT INTO Treasures
  (id, name. gp)
VALUES
  (6, '+1 sword', 1250);
```

```
CREATE (t:Treasure {
  id: 6, gp: 1250 })
SET t.name = '+1 sword'
```

redis

The Relational Database So Far...

Rooms

	id	name
	1	Statue Room
	2	Barracks
	3	Armoury

Monsters

	id	name	cr	xp
	4	Alice the Elf	7	350
	5	Bob the Ogre	5	250

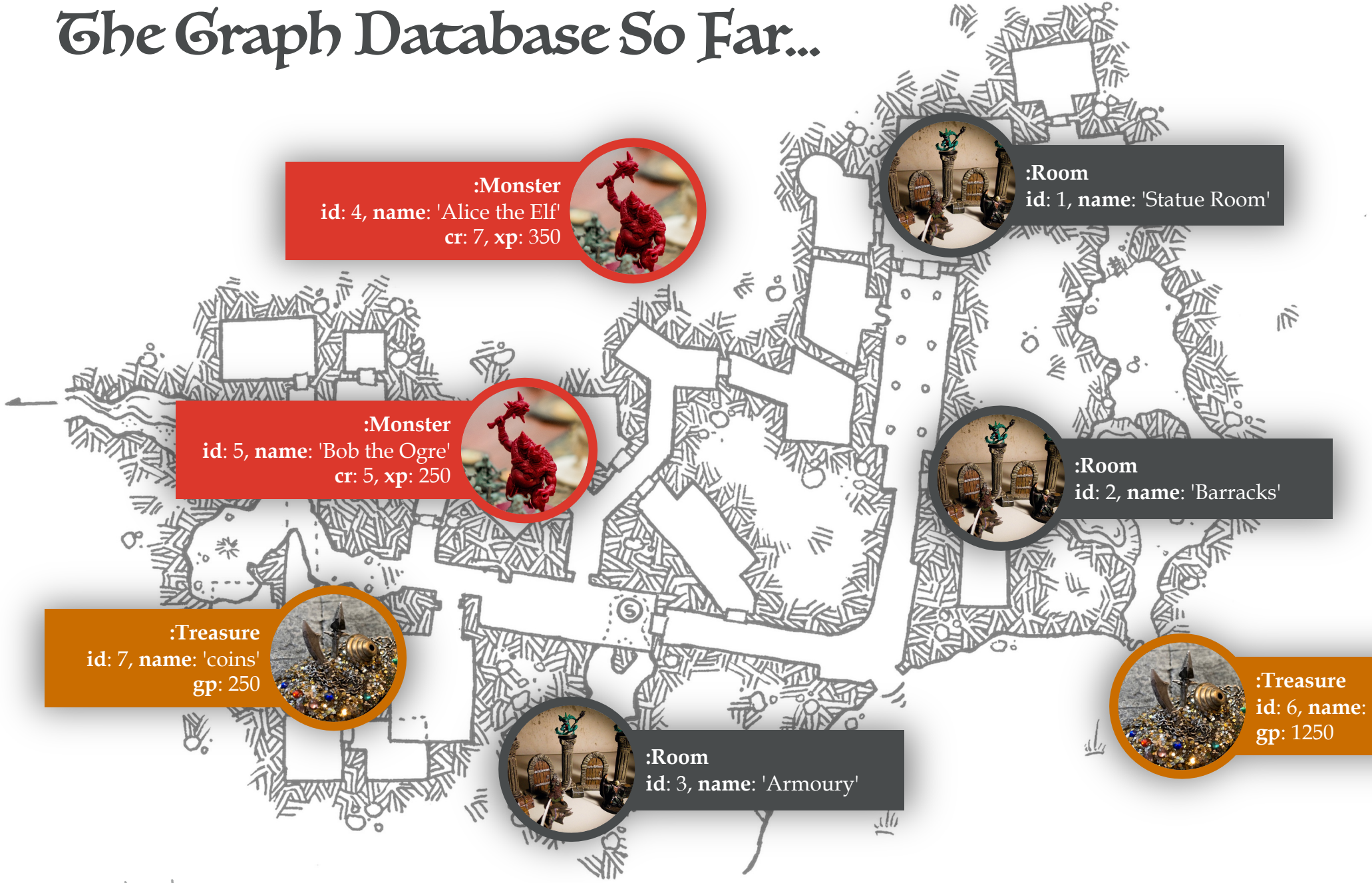
Treasures

	id	name	gp
	6	+1 sword	1250
	7	coins	250

Cartography by Dyson Logos

redis

The Graph Database So Far...



Cartography by Dyson Logos

redis

Putting a Monster in a Room

Rooms		
	id	name
	1	Statue Room
	2	Barracks
	3	Armoury

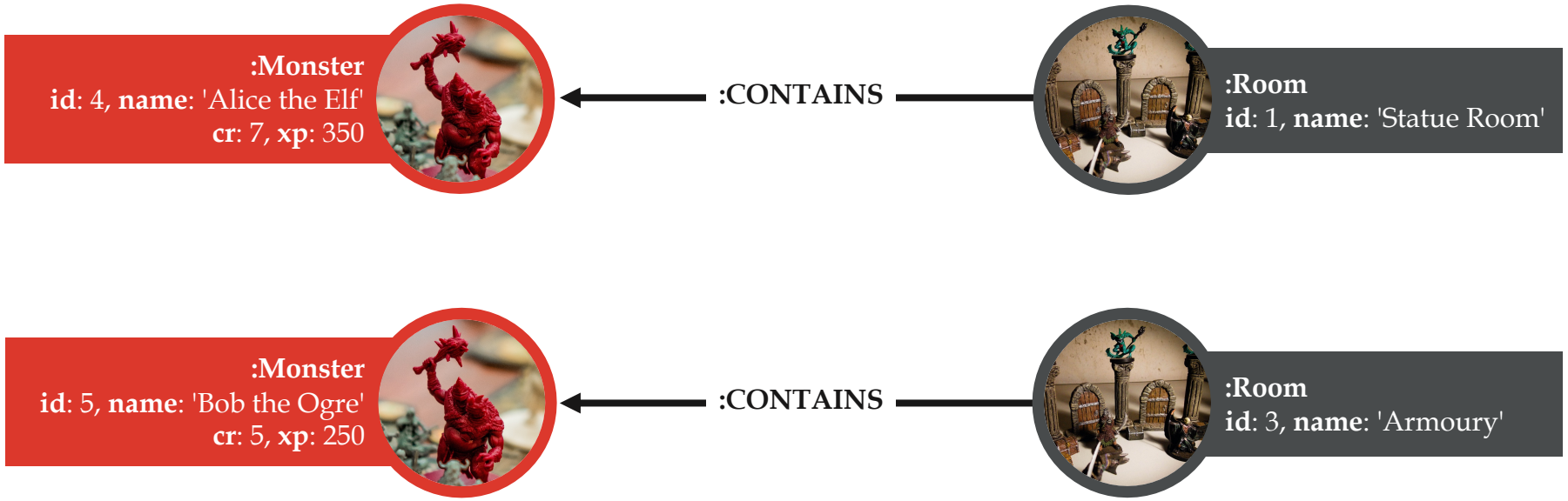
Monsters				
id	name	cr	xp	room_id
4	Alice the Elf	7	350	1
5	Bob the Ogre	5	250	3

```
ALTER TABLE Monsters
ADD room_id int;
```

```
UPDATE Monsters
SET room_id = 1
WHERE id = 4;
```

redis

Putting a Monster in a Room




```
MATCH (r:Room { id: 1 })
MATCH (m:Monster { id: 4 })
CREATE (r)-[:CONTAINS]->(m)
```

redis

Putting Treasure in a Room

Rooms		
	id	name
	1	Statue Room
	2	Barracks
	3	Armoury

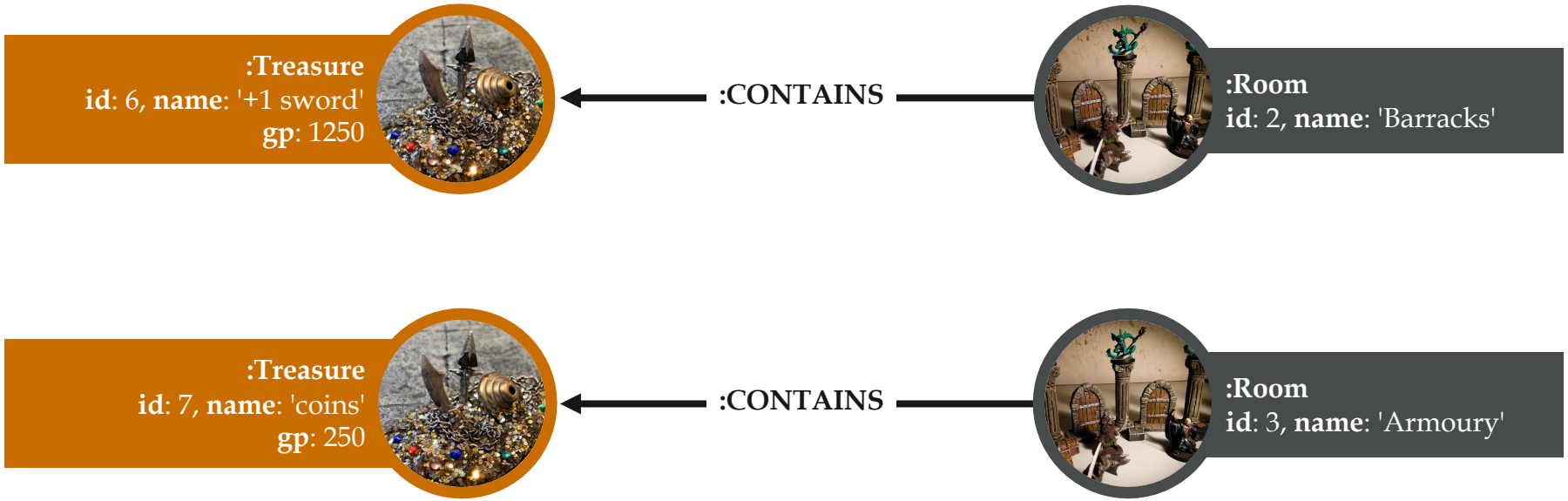
Treasures				
	id	name	gp	room_id
	6	+1 sword	1250	2
	7	coins	250	3

```
ALTER TABLE Treasures
ADD room_id int;
```

```
UPDATE Treasures
SET room_id = 2
WHERE id = 6;
```

redis

Putting Treasure in a Room




```
MATCH (r:Room { id: 2 })
MATCH (t:Treasure { id: 6 })
CREATE (r)-[:CONTAINS]->(t)
```

redis

The Relational Database So Far...

Rooms		
	id	name
	1	Statue Room
	2	Barracks
	3	Armoury

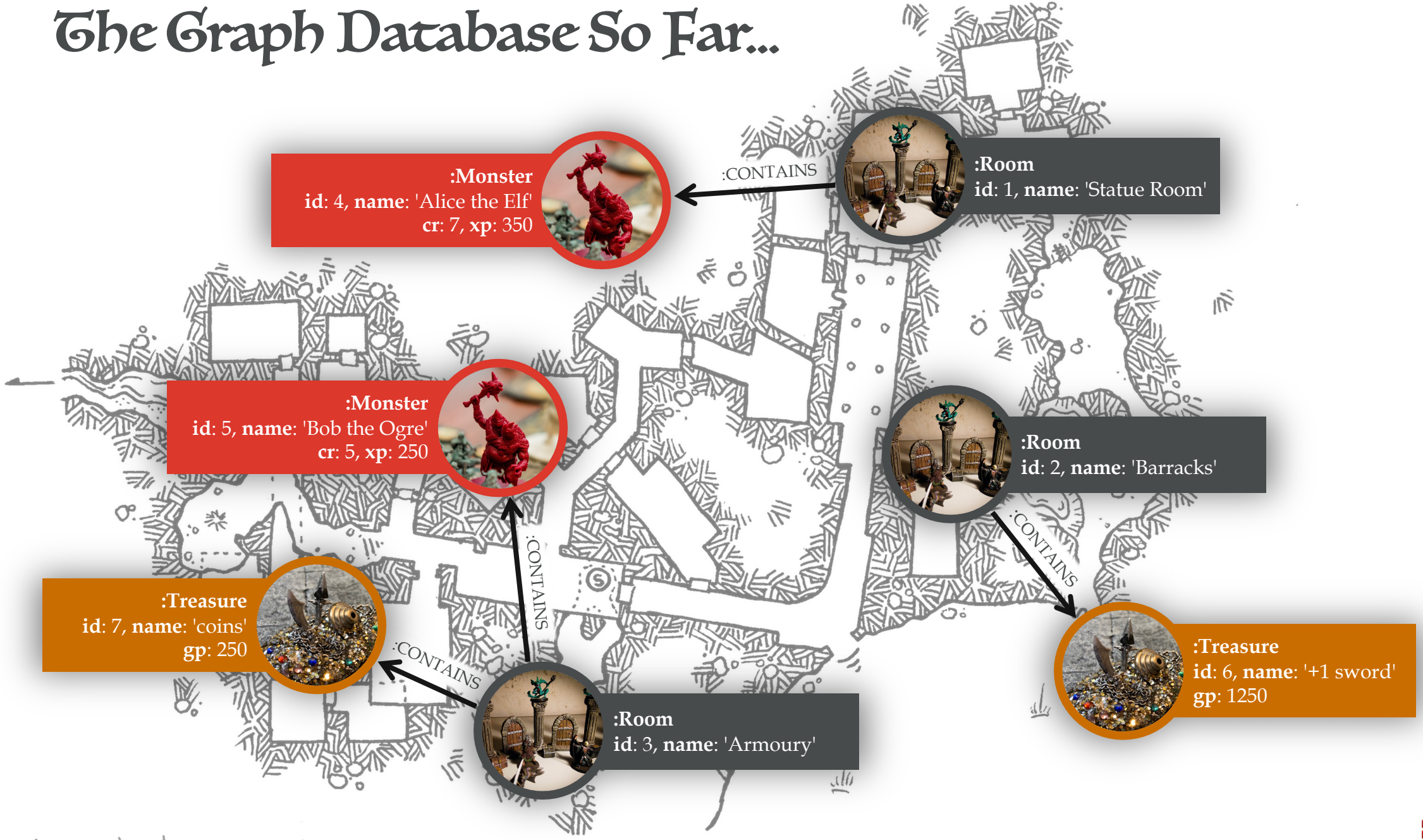
Treasures			
id	name	gp	room_id
6	+1 sword	1250	2
7	coins	250	3

Monsters				
id	name	cr	xp	room_id
4	Alice the Elf	7	350	1
5	Bob the Ogre	5	250	2

Cartography by Dyson Logos

redis

The Graph Database So Far...



Cartography by Dyson Logos

redis

Munchkin Time!

Cartography by Dyson Logos

redis

Farming All the XP!

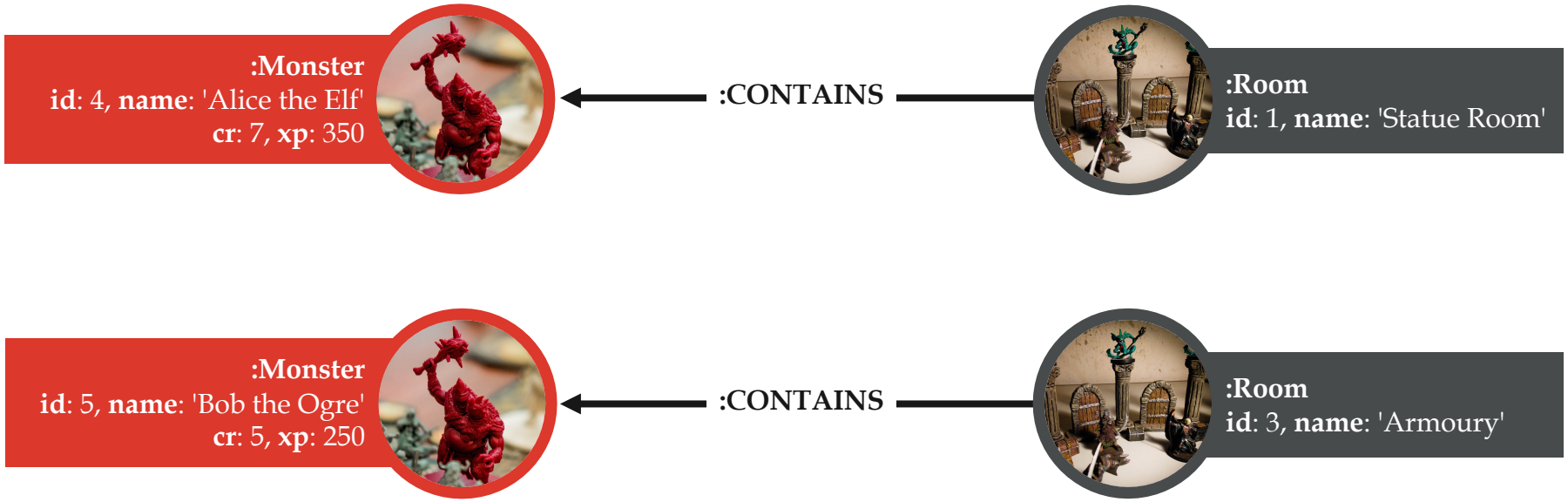
Rooms		
	id	name
	1	Statue Room
	2	Barracks
	3	Armoury

Monsters				
	id	name	cr	xp	room_id
	4	Alice the Elf	7	350	1
	5	Bob the Ogre	5	250	3

```
SELECT
  r.id, r.name, m.xp
FROM
  Rooms as r,
  Monsters as m
WHERE
  r.id = m.room_id
ORDER BY
  m.xp DESC;
```

redis

Farming All the XP!




```
MATCH (r:Room)-[:CONTAINS]->(m:Monster)
RETURN r.id, r.name, m.xp
ORDER BY m.xp DESC
```

redis

Getting All the Gold!

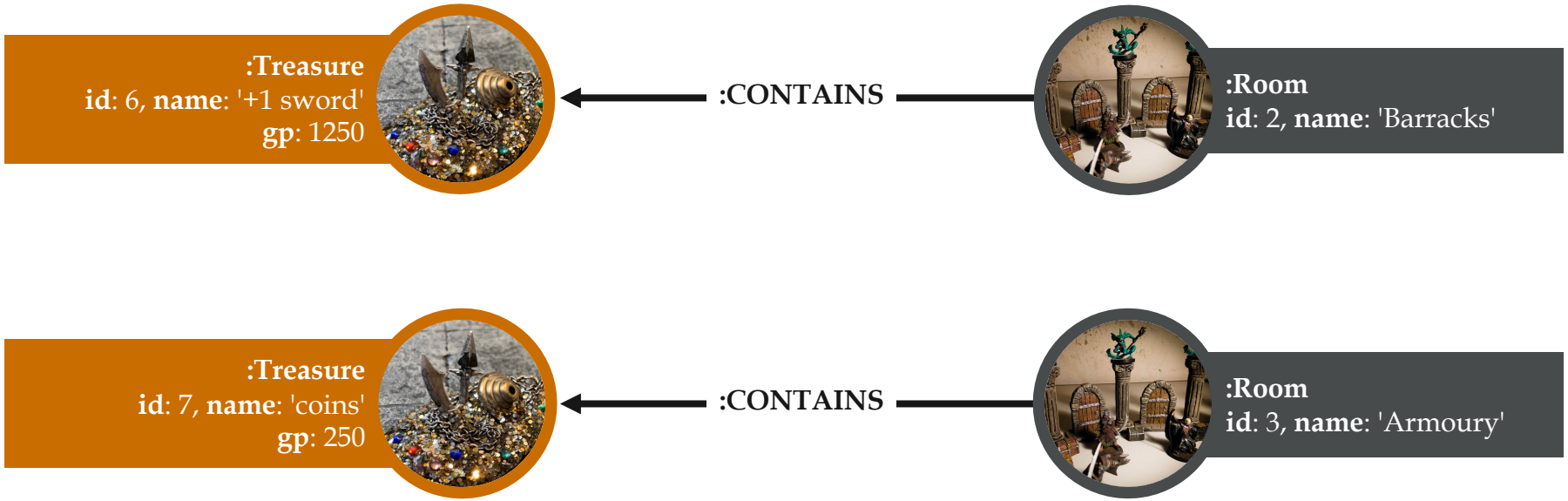
Rooms		
	id	name
	1	Statue Room
	2	Barracks
	3	Armoury

Treasures			
	id	name	gp	room_id
	6	+1 sword	1250	2
	7	coins	250	3

```
SELECT
  r.id, r.name, t.gp
FROM
  Rooms as r,
  Treasures as t
WHERE
  r.id = t.room_id
ORDER BY
  t.gp DESC;
```

redis

Getting All the Gold!




```
MATCH (r:Room)-[:CONTAINS]->(t:Treasure)
RETURN r.id, r.name, t.gp
ORDER BY t.gp DESC
```

redis

So Far, So Good... So What?

Cartography by Dyson Logos

redis

Connecting Rooms

Rooms	
id	name
1	Statue Room
2	Barracks
3	Armoury

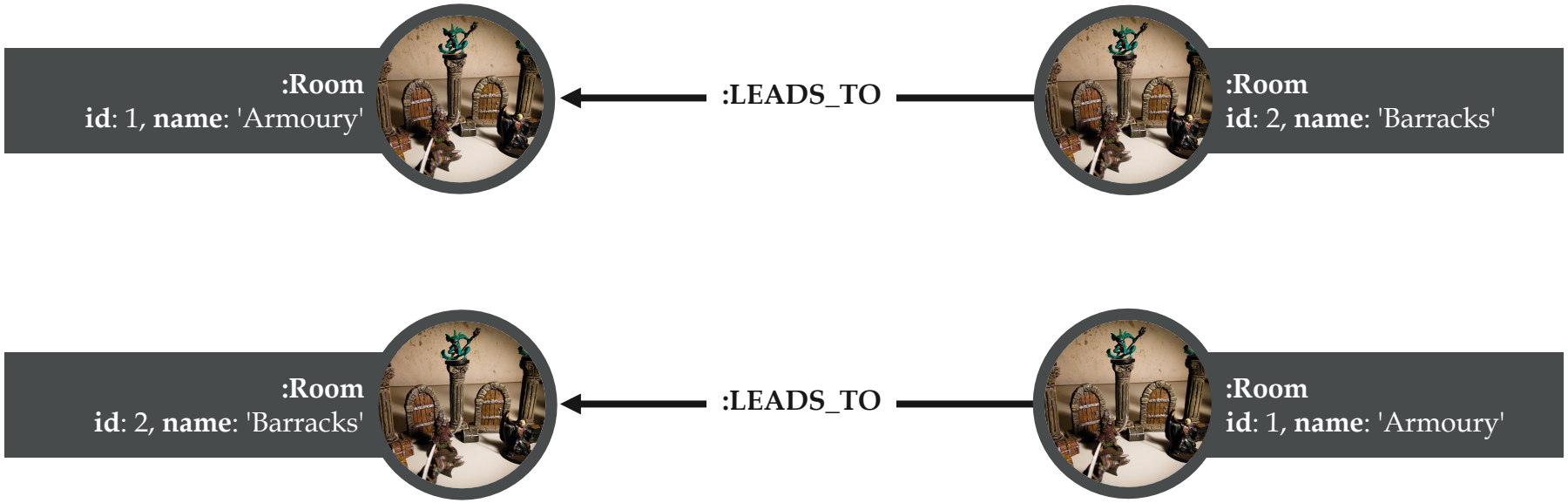
Connections	
from_id	to_id
1	1
1	2
1	3
2	1
2	3
3	1

```
CREATE TABLE Connections (
  from_id int,
  to_id int
);
```

```
INSERT INTO Connections
  (from_id, to_id)
VALUES
  (1, 2);
```

redis

Connecting the Rooms




```
MATCH (r1:Room { id: 1 })
MATCH (r2:Room { id: 2 })
CREATE (r1)-[:LEADS_TO]->(r2)
```

redis

The Relational Database

Rooms

	id	name
	1	Statue Room
	2	Barracks
	3	Armoury

Connections

	from_id	to_id
	1	1
	1	2
	1	3
	2	1
	2	3
	3	1

Treasures

	id	name	gp	room_id
	6	+1 sword	1250	2
	7	coins	250	3

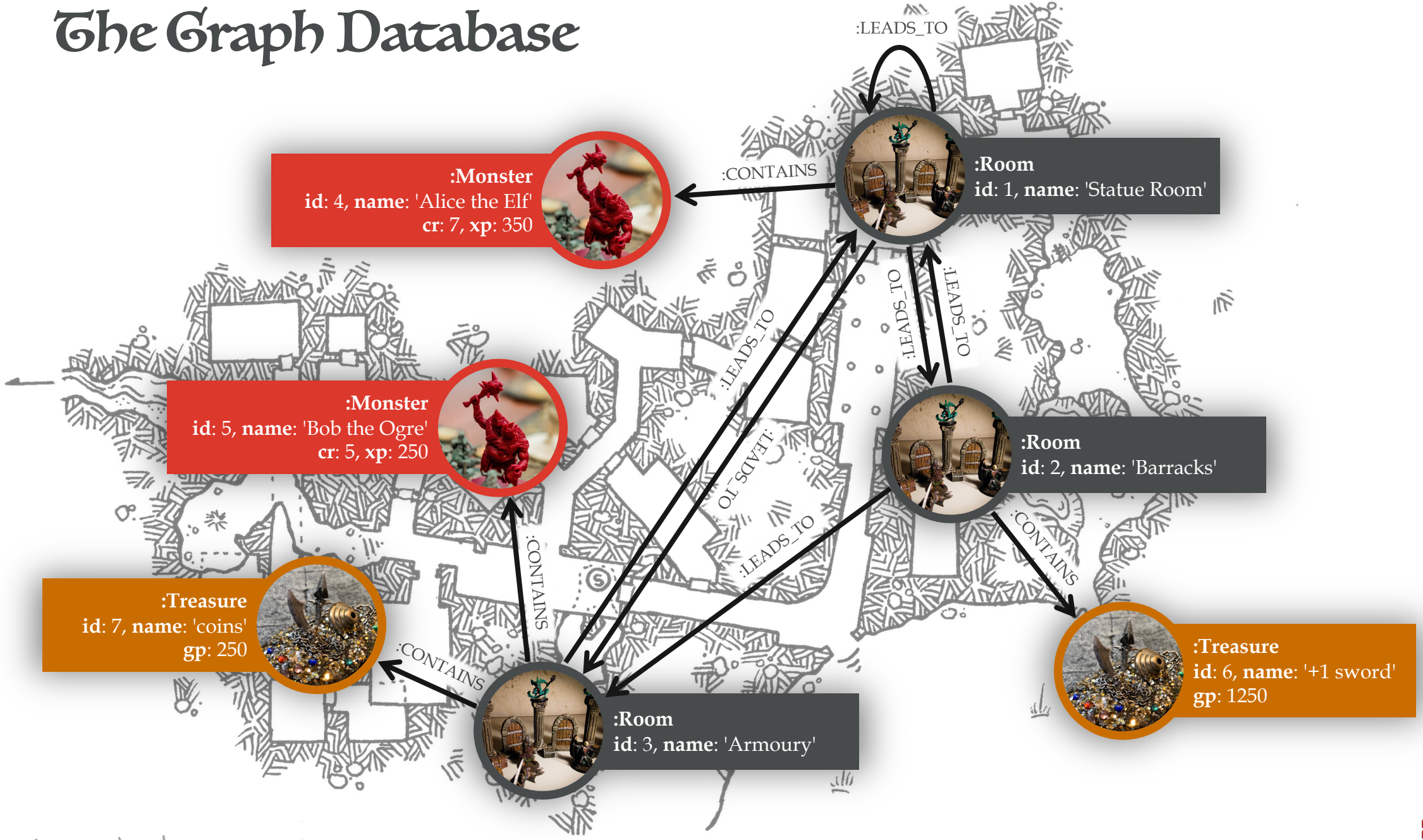
Monsters

	id	name	cr	xp	room_id
	4	Alice the Elf	7	350	1
	5	Bob the Ogre	5	250	2

Cartography by Dyson Logos

redis

The Graph Database



Cartography by Dyson Logos

redis

Super Munchkin Time!

Cartography by Dyson Logos

redis

Variable Length Relationships


```
(r1:Room)-[:LEADS_TO*]->(r2:Room)
```

matches room node

matches multiple leads_to relationships

matches room node

redis

Variable Length Relationships

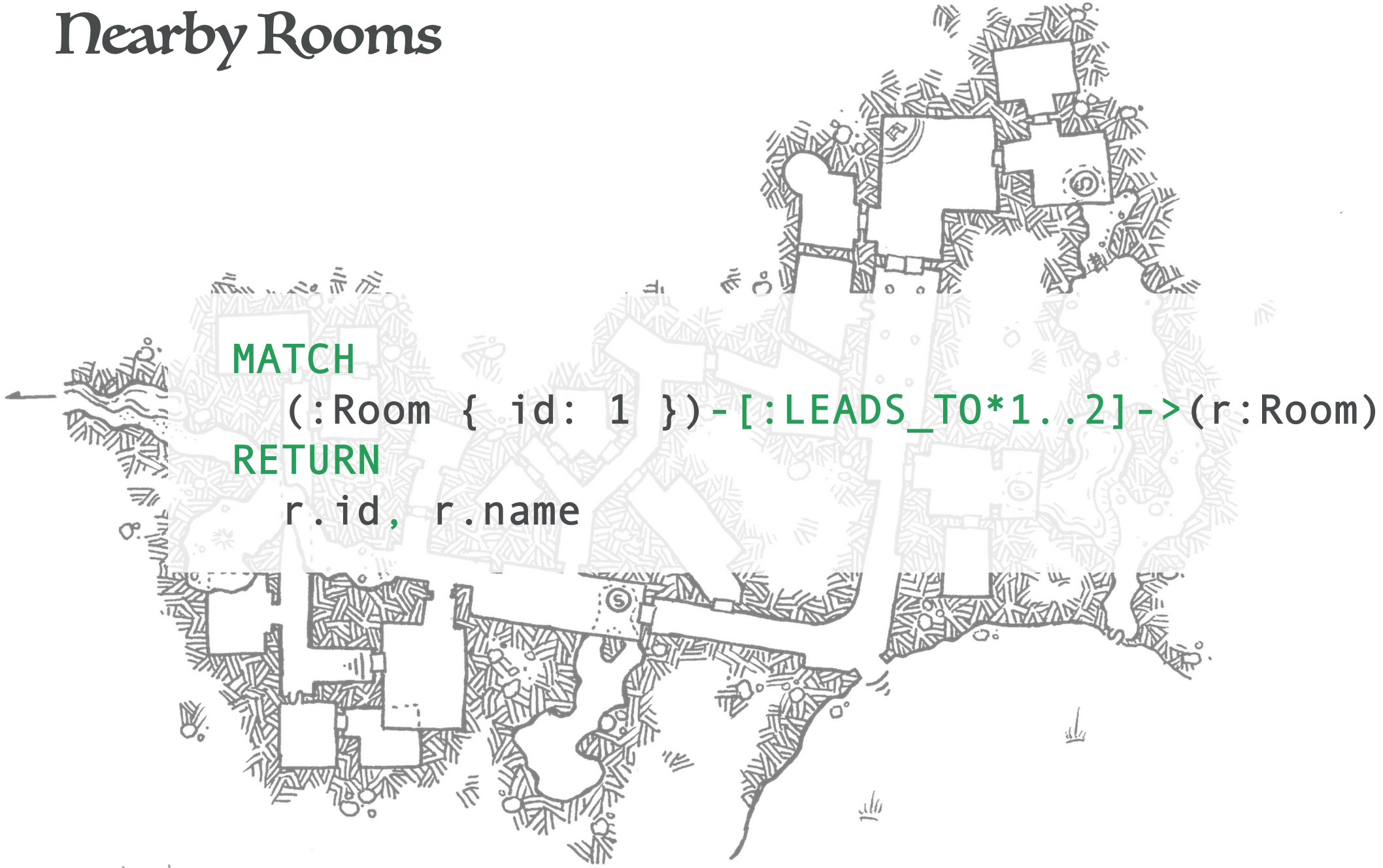

```
(r1:Room)-[:LEADS_TO*1..3]->(r2:Room)
```

minimum hops

maximum hops

redis

Nearby Rooms

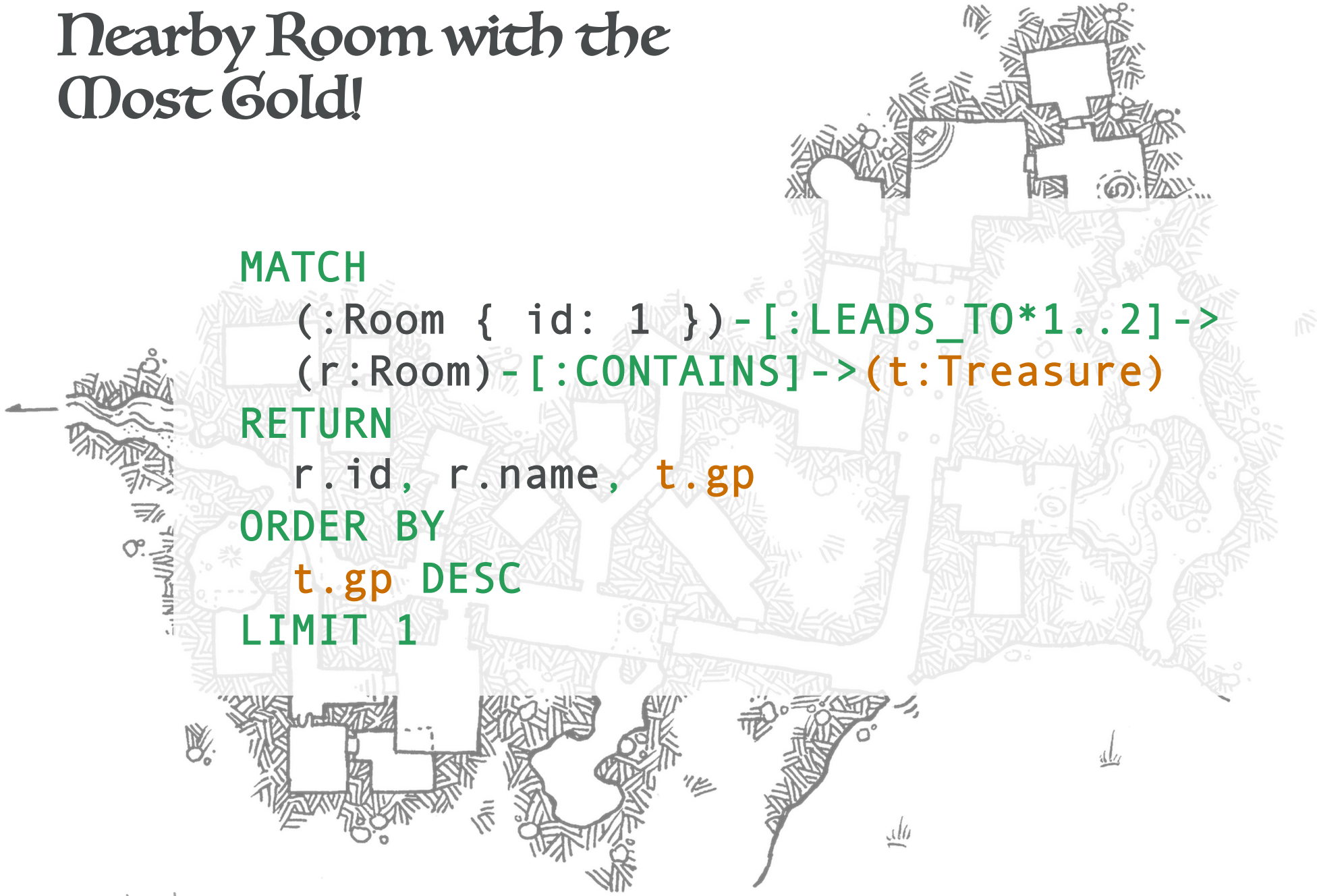


```
MATCH
    (:Room { id: 1 })-[:LEADS_TO*1..2]->(r:Room)
RETURN
    r.id, r.name
```

Cartography by Dyson Logos

redis

Nearby Room with the Most Gold!

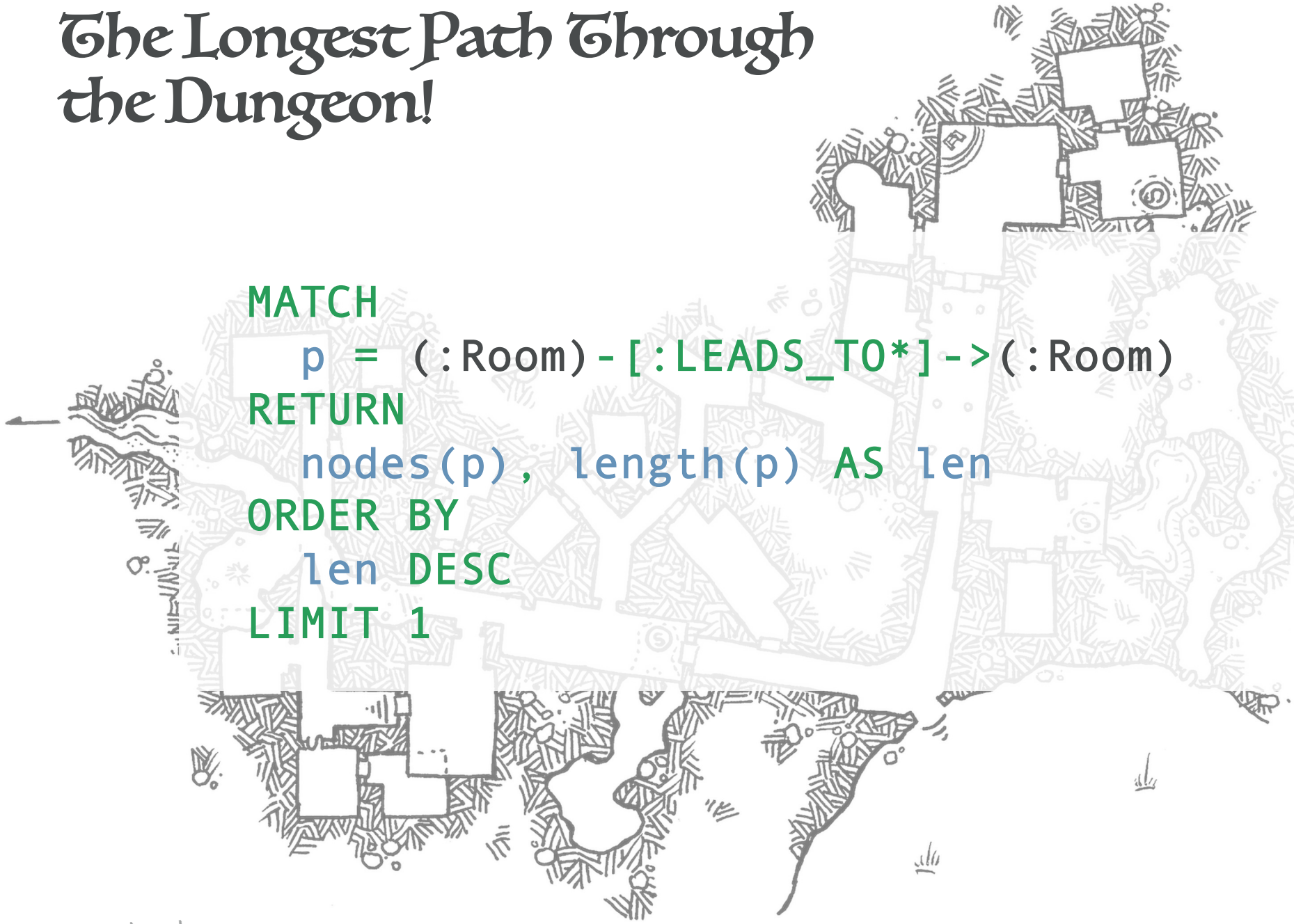


```
MATCH
   (:Room { id: 1 })-[:LEADS_TO*1..2]->
   (r:Room)-[:CONTAINS]->(t:Treasure)
RETURN
   r.id, r.name, t.gp
ORDER BY
   t.gp DESC
LIMIT 1
```

Cartography by Dyson Logos

redis

The Longest Path Through the Dungeon!

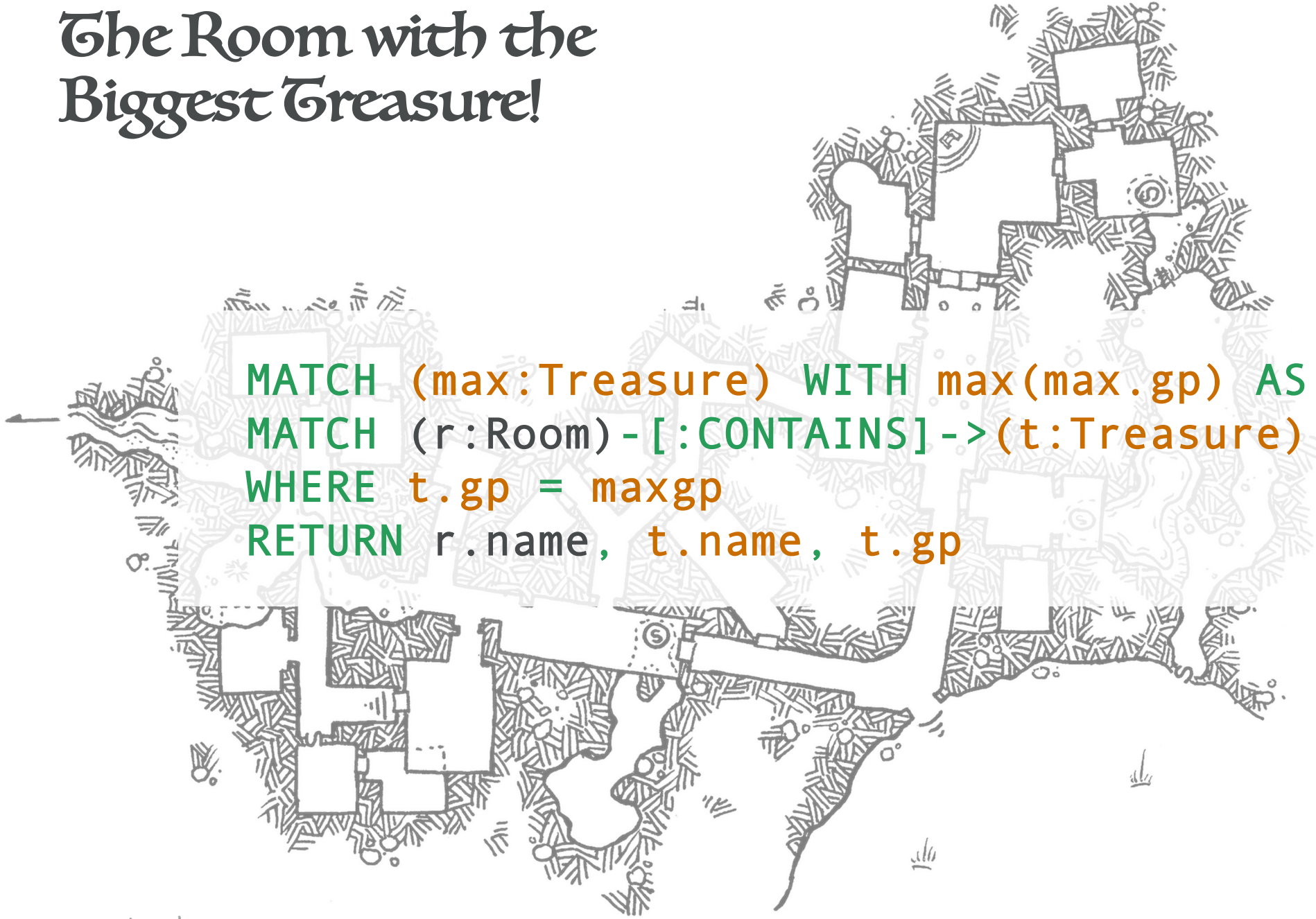


```
MATCH
   p = (:Room)-[:LEADS_TO*]->(:Room)
RETURN
   nodes(p), length(p) AS len
ORDER BY
   len DESC
LIMIT 1
```

Cartography by Dyson Logos

redis

The Room with the Biggest Treasure!

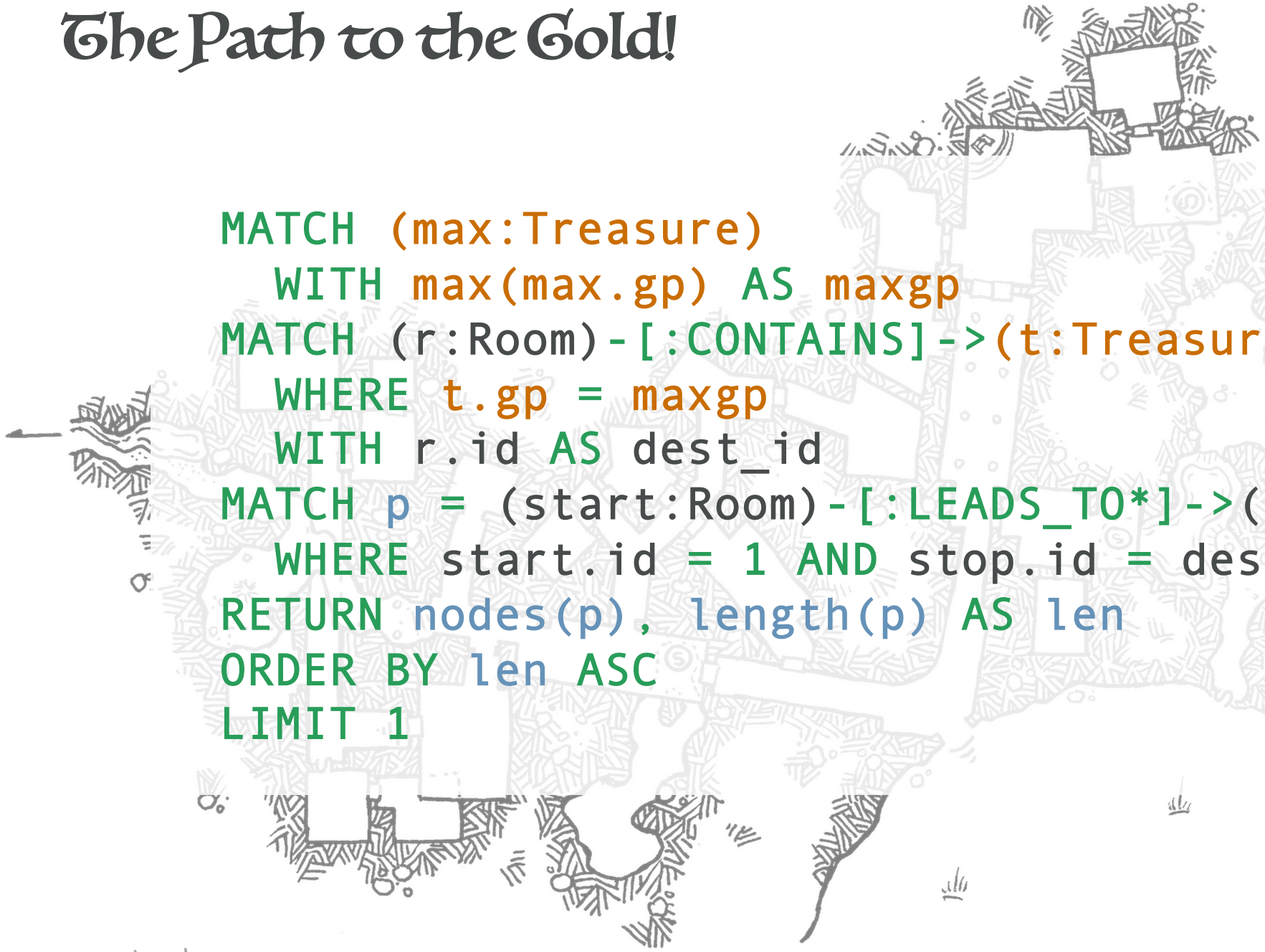


```
MATCH (max:Treasure) WITH max(max.gp) AS maxgp
MATCH (r:Room)-[:CONTAINS]->(t:Treasure)
WHERE t.gp = maxgp
RETURN r.name, t.name, t.gp
```

Cartography by Dyson Logos

redis

The Path to the Gold!

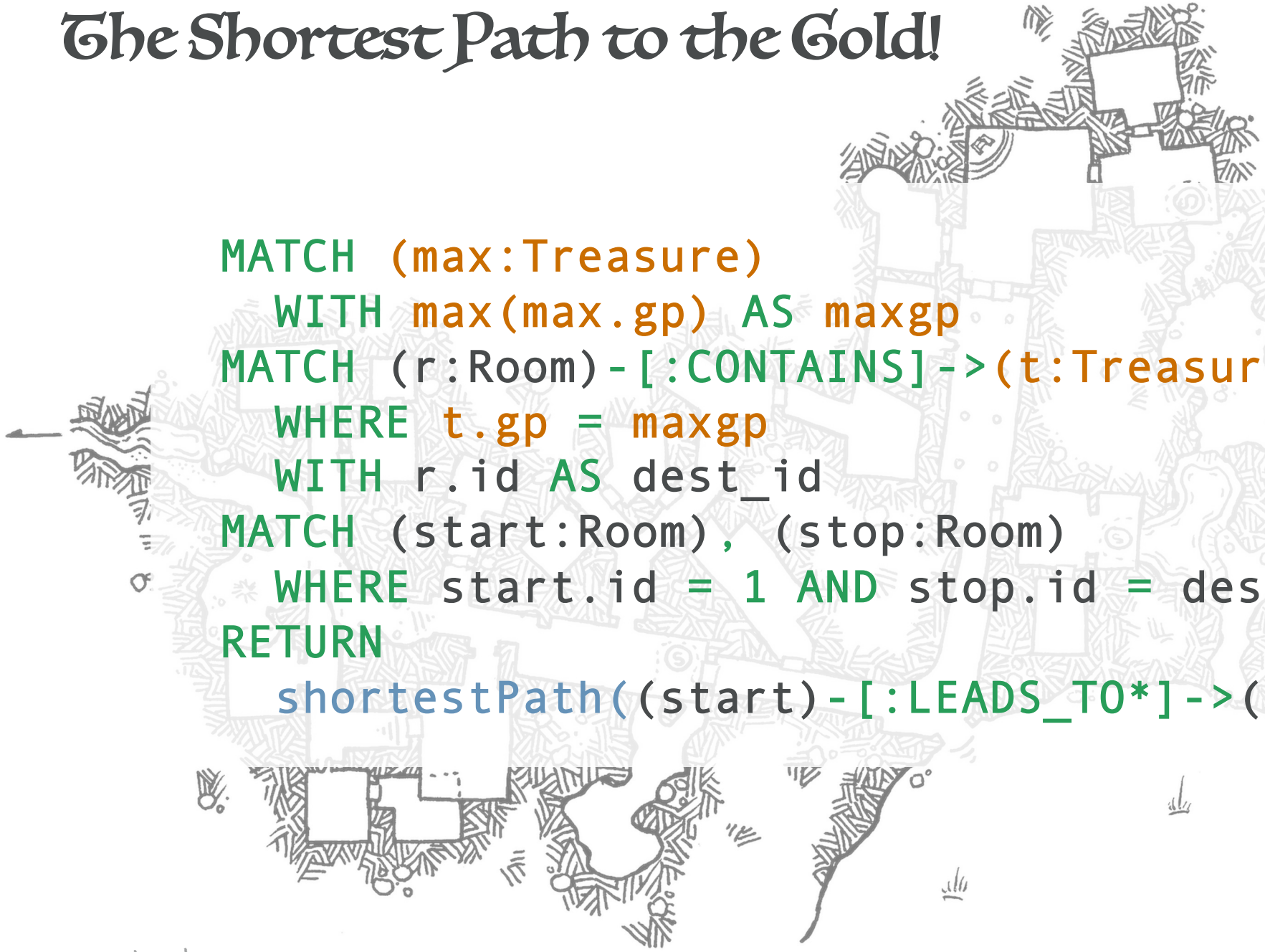


```
MATCH (max:Treasure)
  WITH max(max.gp) AS maxgp
MATCH (r:Room)-[:CONTAINS]->(t:Treasure)
  WHERE t.gp = maxgp
  WITH r.id AS dest_id
MATCH p = (start:Room)-[:LEADS_TO*]->(stop:Room)
  WHERE start.id = 1 AND stop.id = dest_id
RETURN nodes(p), length(p) AS len
ORDER BY len ASC
LIMIT 1
```

Cartography by Dyson Logos

redis

The Shortest Path to the Gold!



```
MATCH (max:Treasure)
  WITH max(max.gp) AS maxgp
MATCH (r:Room)-[:CONTAINS]->(t:Treasure)
  WHERE t.gp = maxgp
  WITH r.id AS dest_id
MATCH (start:Room), (stop:Room)
  WHERE start.id = 1 AND stop.id = dest_id
RETURN
  shortestPath((start)-[:LEADS_TO*]->(stop))
```

Cartography by Dyson Logos

redis


Demo

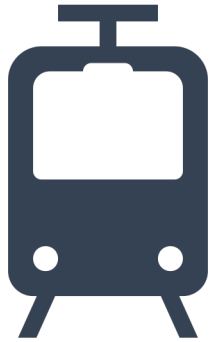
Practical Applications

Social Networks—find friends of a friend and a friend of your friends

Genealogy—querying descendants, find who is related to whom

Transportation Networks—optimizing routes for drivers and mass transit riders

Logistics—finding the source of goods for a store; finding all the stores a source ultimately distributes to



Redis RedisJSON RediSearch RedisGraph RedisTimeSeries RedisBloom

redis.io/docs/stack/

I've Been Working for Redis for 1d4-1 Years

Redis Discord

https://discord.gg/redis

Redis Insight

https://redis.com/redisinsight

Redis Cloud

https://redis.com/try-free/

redis

Code, Slides, and 3d6+2 Pull Requests

https://github.com/guyroyse/dnd-and-graph-databases

Guy Royse

Developer Advocate

redis

@guyroyse

github.com/guyroyse

guy.dev


Thanks!
redis