Graph Databases

Christopher Thielen, Lead Application Developer, DSS IT

The Breadth of the NoSQL Movement

Relationships Are Data

Join Table as a hack.

Graph Concepts

- Nodes
- Edges
- Properties
- Relationship complexity grows linearly, not exponentially

Cypher vs SQL: Schema creation

```
CREATE TABLE `students` (
  `id` int(11) NOT NULL,
  `name` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id`)
)
```

```
Not needed.
```

Cypher vs SQL: Insert data

```
INSERT INTO students (`name`) values
('Christopher');
```

```
CREATE (n:Student {name: 'Christopher'})
```

Cypher vs SQL: Query all students

```
SELECT * FROM students;
```

```
MATCH (n:Student) RETURN n
```

Cypher vs SQL: Enroll a student

```
INSERT INTO enrollments (`student_id`,
`class_id`) values (1, 5);
```

```
MATCH (s:Student),(c:Class)
WHERE s.name = 'Christopher' AND c.name =
'From SQL to NoSQL'
CREATE (s)-[r:ENROLLED_IN]->(c)
RETURN type(r)
```

Cypher vs SQL: Class roster

```
SELECT * FROM students
LEFT JOIN enrollments ON students.id =
enrollments.student_id
LEFT JOIN classes ON classes.id =
enrollments.class_id
WHERE classes.title = 'From SQL to NoSQL';
```

```
MATCH (s:Student)-[:ENROLLED_IN]-
>(c:Class{name:'From SQL to NoSQL'}) RETURN
s, c
```

Cypher vs SQL: Recommendations

```
??? (Yes, it is possible.)
```

```
MATCH (Class{name:'From SQL to NoSQL'})<-
[:ENROLLED_IN]-(Student)-[:ENROLLED_IN]-
>(c) RETURN c, count(c)
```

Demo