



UNIT 0x0F

NoSQL

NoSQL = Not only SQL...

Idee

- NoSQL bezeichnet Typen von DBMS, die einen nicht-relationalen Ansatz verfolgen
- ER-Modellierung nach wie vor sinnvoll

Q&A

- Vor- bzw. Nachteile relationaler DBMS?
- Wo liegen ggf. Schwierigkeiten bei Alternativen?

Überblick

NoSQL Typen (unvollständig)

- Graphendatenbanken
- Objektorientierte Datenbanken
- Key-Value Datenbanken
- Dokumentenorientierte Datenbanken

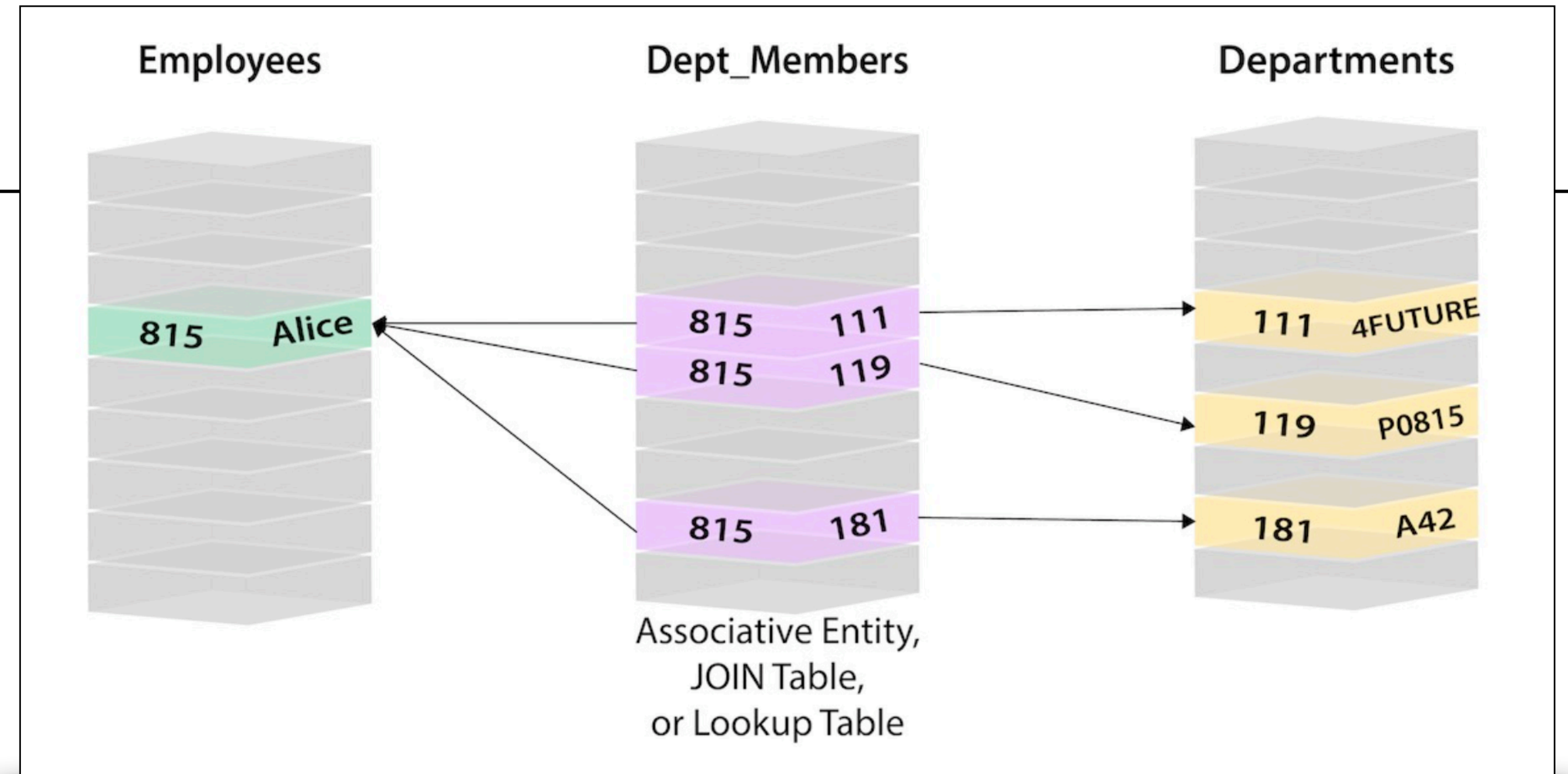
Abgrenzung und Bezeichnung

- unscharf
- relationale DBMS bieten z.T. obige Tabellen- oder Spaltentypen an, beispielsweise unterstützt MySQL nativ JSON Daten (Dokumente) und implementiert einen effizienten Zugriff auf einzelne Attribute innerhalb eines solchen Datums

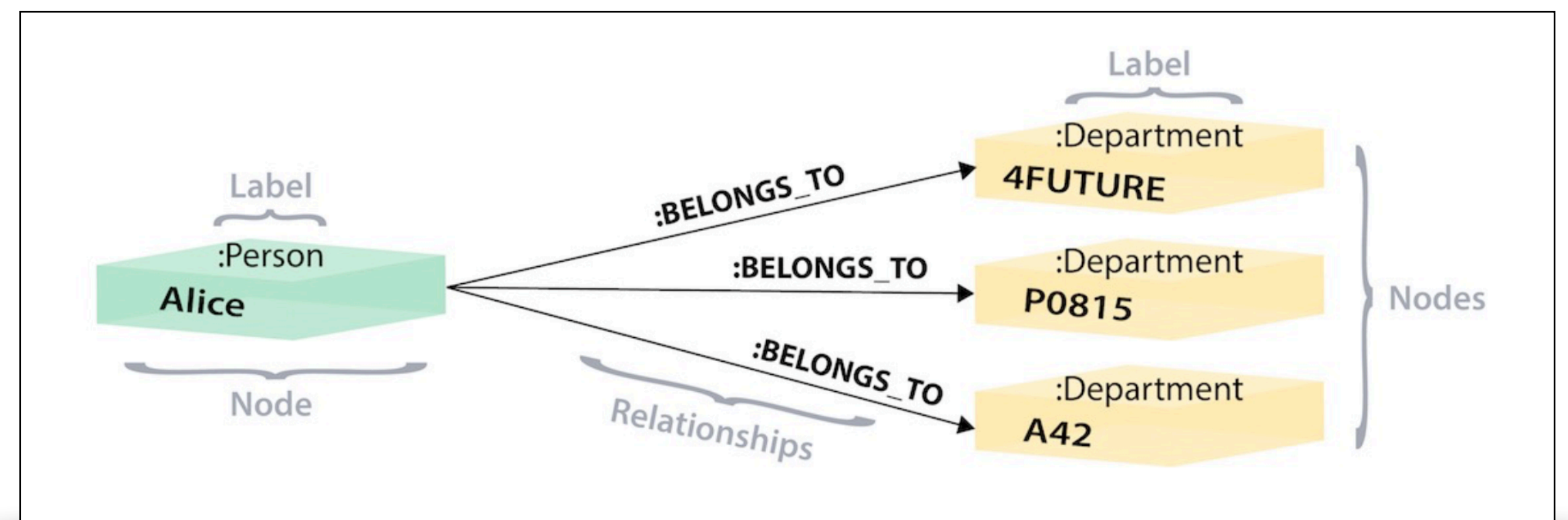
NoSQL Typen

Graphendatenbanken

- Stark vernetzte Daten (Netzwerk) können hier inkl. der Eigenschaften von Knoten und Kanten abgelegt werden
- Spezielle Variante: Triplestore (Bob kennt Fred), z.B. im Forschungsdatenmanagement
- Bsp. neo4j



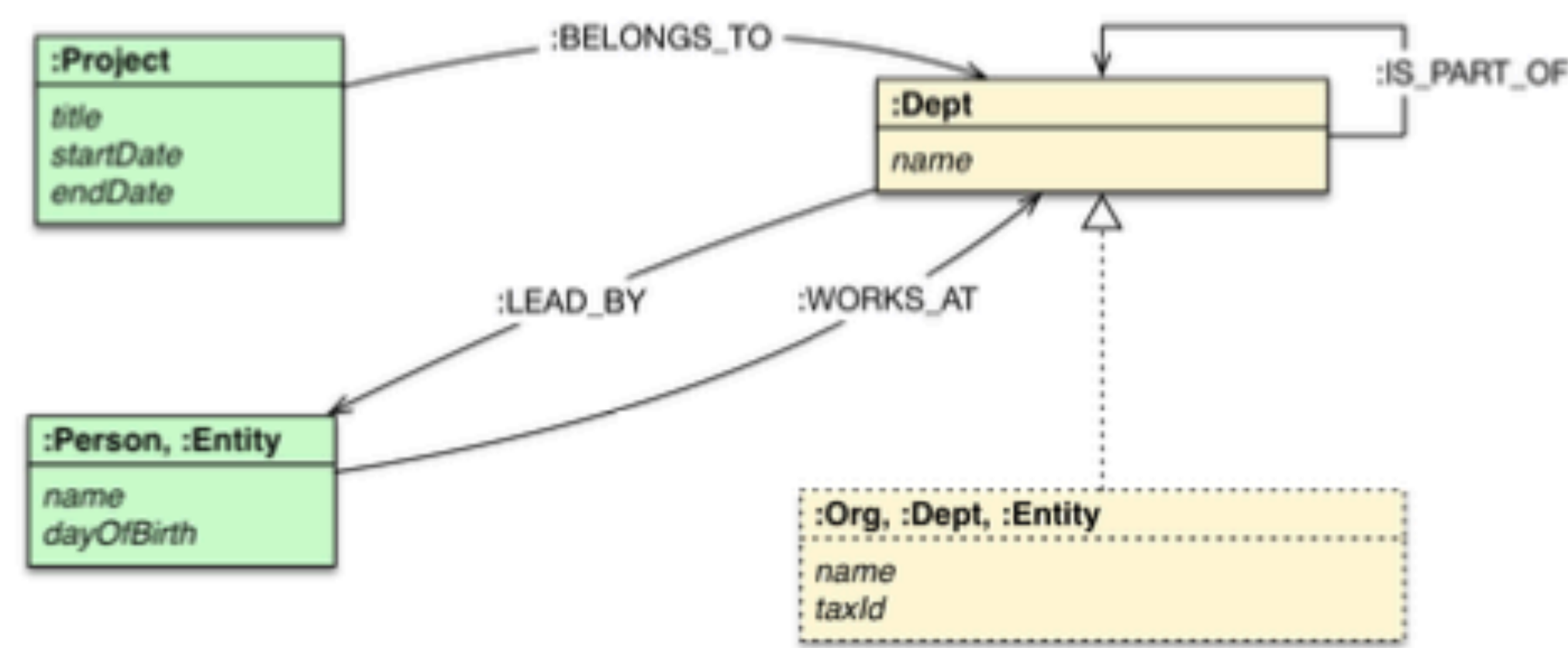
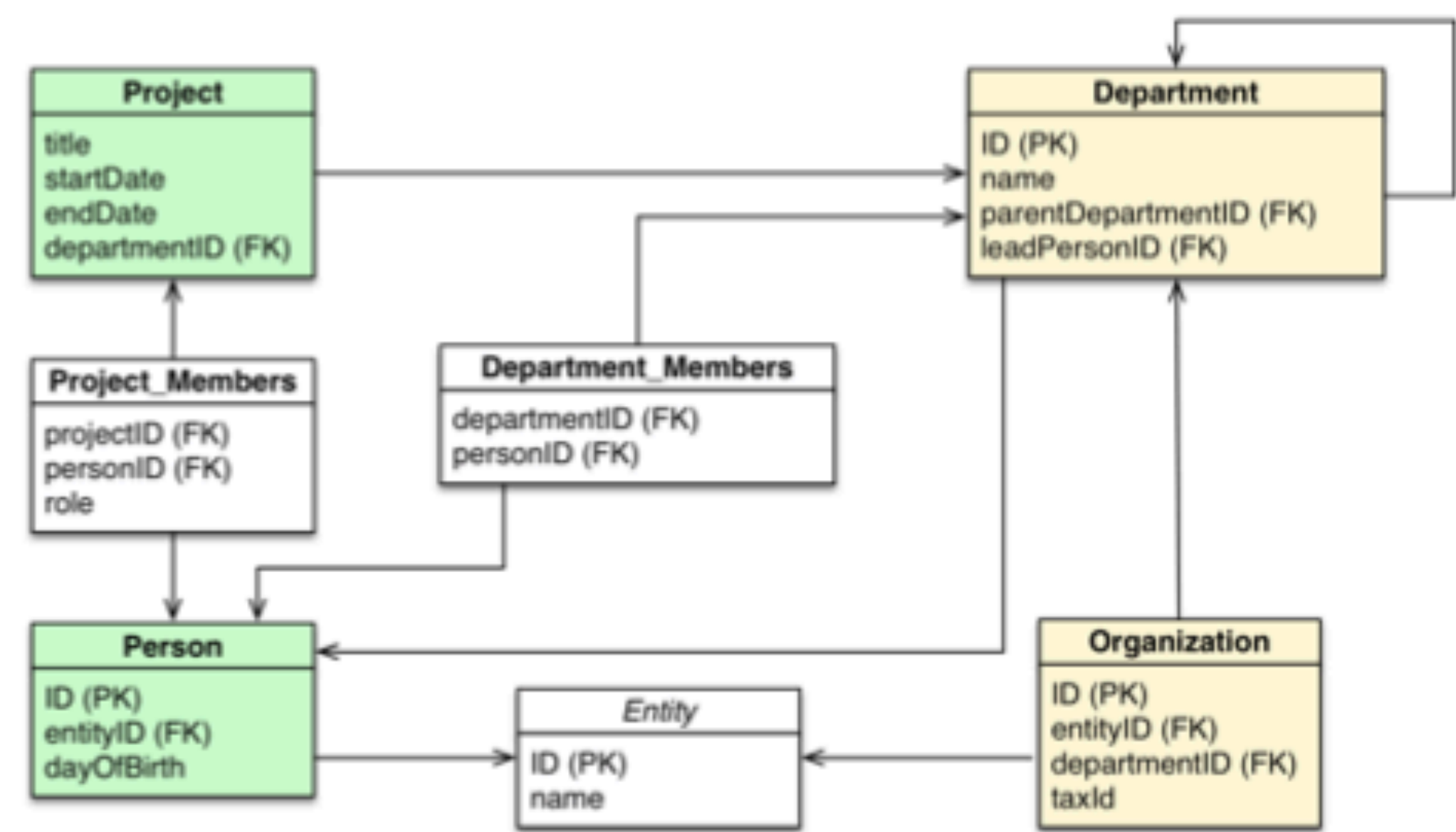
Rel. Sicht, Quelle neo4j



Nodes, Quelle neo4j

NoSQL Typen

Graphendatenbanken neo4j



Quelle neo4j

NoSQL Typen

Graphdatenbanken

SQL Statement Sql	Cypher Statement Cypher
<pre>SELECT name FROM Person LEFT JOIN Person_Department ON Person.Id = Person_Department.PersonId LEFT JOIN Department ON Department.Id = Person_Department.DepartmentId WHERE Department.name = "IT Department"</pre>	<pre>MATCH (p:Person)-[:WORKS_AT]->(d:Dept) WHERE d.name = "IT Department" RETURN p.name</pre>

Quelle neo4j

NoSQL Typen

Objektorientierte Datenbanken (ODBMS)

- Verwaltung komplexer Objekte
- Unterstützung von Objektidentitäten
- Kapselung der Objekte entsprechend objektorientierter Programmierung
- Klassen sind in Klassenhierarchien angeordnet
- ODBMS stellt Manipulationssprache bereit
- Bsp. ZooDB
 - implementiert Apache JDO
- Bsp. db4o
 - unterstützt .NET und begrenzt LINQ
 - keine Unterstützung mehr seit 2014 ...

NoSQL Typen

Objektorientierte Datenbanken (ODBMS)

- Sehr kleine Bibliothek.
- Objekte können gespeichert werden ohne Metainformationen, z.B. Klasseninformationen, anzugeben.

```
// Öffne eine Datenbank
ObjectContainer db = Db4o.openFile("C:/beispiel.db");
try {
    // Speichere zwei Personen
    db.store(new Person("John Doe", "TOP INC", 45));
    db.store(new Person("Clara Himmel", "BB Radio", 30));
    // Iteriere über alle Personen
    ObjectSet result = db.queryByExample(new Person());
    while (result.hasNext()) {
        System.out.println(result.next());
    }
    // Verändere eine Person
    result = db.queryByExample(new Person("Clara Himmel"));
    Person found = (Person) result.next();
    found.setAge(25);
    db.store(found);
    // Lösche eine Person
    db.delete(found);
    // Schreibe die Änderungen fest
    db.commit();
}
finally {
    // Schließe die Datenbank
    db.close();
}
```

db4o

NoSQL Typen

Key-Value Datenbanken

- In Key-Value-Datenbanken werden Werte (Value) über einen Schlüssel (Key) eindeutig identifiziert und abgelegt.
- Werte bestehen aus Tupeln, Relationen oder Dokumenten.
- Trennung Key-Value- und Dokument-Datenbanken unscharf.
- Beispiel Google BigTable
 - baut auf Google File System auf (große Files, verteilt),
 - Anwendungen sind Google Maps, Google Bücher, YouTube, Google Earth oder Google Cloud Datastore als Teil der Google Cloud Platform.
- Beispiel Amazon Dynamo
 - Cloud Datenbank,
 - unterstützt auch Dokument- und Key-Value Datenbanken.

NoSQL Typen

Dokumentenorientierte Datenbanken

- Dokumente bilden Grundlage der Speicherung.
- Dokumente sind eindeutig identifizierbar, bestehen aus Key-Value-Paaren mit frei festlegbarem Schema.
- Beispiel CouchDB.
- Beispiel MongoDB.

NoSQL Typen

Dokumentenorientierte Datenbanken CouchDB

- Verwendet für Webseiten oder in Facebook-Anwendungen, in Ubuntu z.B. für Adressen und Lesezeichen.
- Speichert Daten als Dokumente, die aus JSON-Objekten bestehen (früher XML).
- Relationen können über eindeutige ids zwischen Dokumenten hergestellt werden
→ daher sind ER-Modelle auch modellierbar.

```
{
  "_id": "biking",
  "_rev": "AE19EBC7654",

  "title": "Biking",
  "body": "My biggest hobby is mountain biking",
  "date": "2009/01/30 18:04:11"
}
```

```
{
  "_id": "bought-a-cat",
  "_rev": "4A3BBEE711",

  "title": "Bought a Cat",
  "body": "I went to the the pet store",
  "date": "2009/02/17 21:13:39"
}
```

```
{
  "_id": "hello-world",
  "_rev": "43FBA4E7AB",

  "title": "Hello World",
  "body": "Well hello and welcome to my blog",
  "date": "2009/01/15 15:52:20"
}
```

NoSQL Typen

Dokumentenorientierte Datenbanken CouchDB

- Queries via “views” (function) und map-reduce-functions.
- Ergebnisse in B-Bäume.

```
function(doc) {  
  if(doc.date && doc.title) {  
    emit(doc.date, doc.title);  
  }  
}
```



```
{  
  "total_rows": 3,  
  "offset": 0,  
  "rows": [  
    {  
      "key": "2009/01/15 15:52:20",  
      "id": "hello-world",  
      "value": "Hello World"  
    },  
    {  
      "key": "2009/01/30 18:04:11",  
      "id": "biking",  
      "value": "Biking"  
    },  
    {  
      "key": "2009/02/17 21:13:39",  
      "id": "bought-a-cat",  
      "value": "Bought a Cat"  
    }  
  ]  
}
```