BI/SQL Server

BI/SQL server support within Stardog

Taught by:

Al Baker
VP, Enterprise Solutions

Learning Objectives

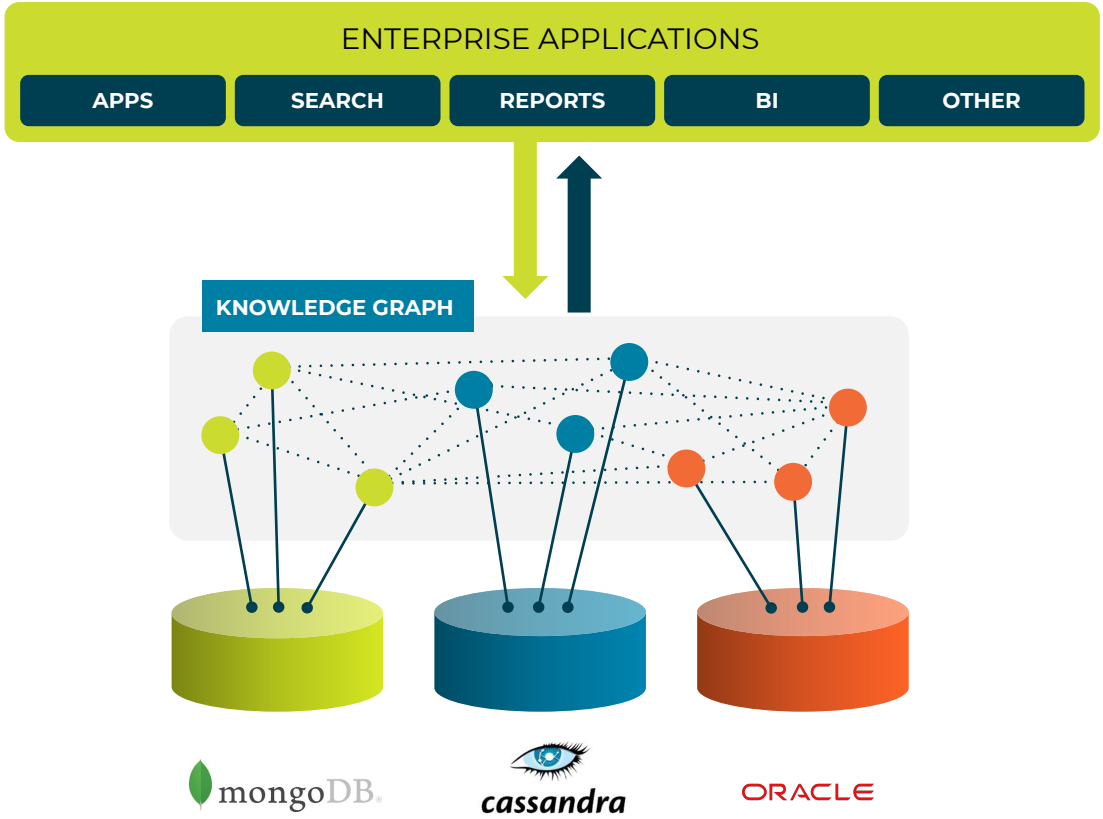
- Understand the Stardog BI/SQL server feature
- Use this feature through various methods
- Implement the concepts of the SQL Mapping Syntax

Overview

STARDOG ACADEMY

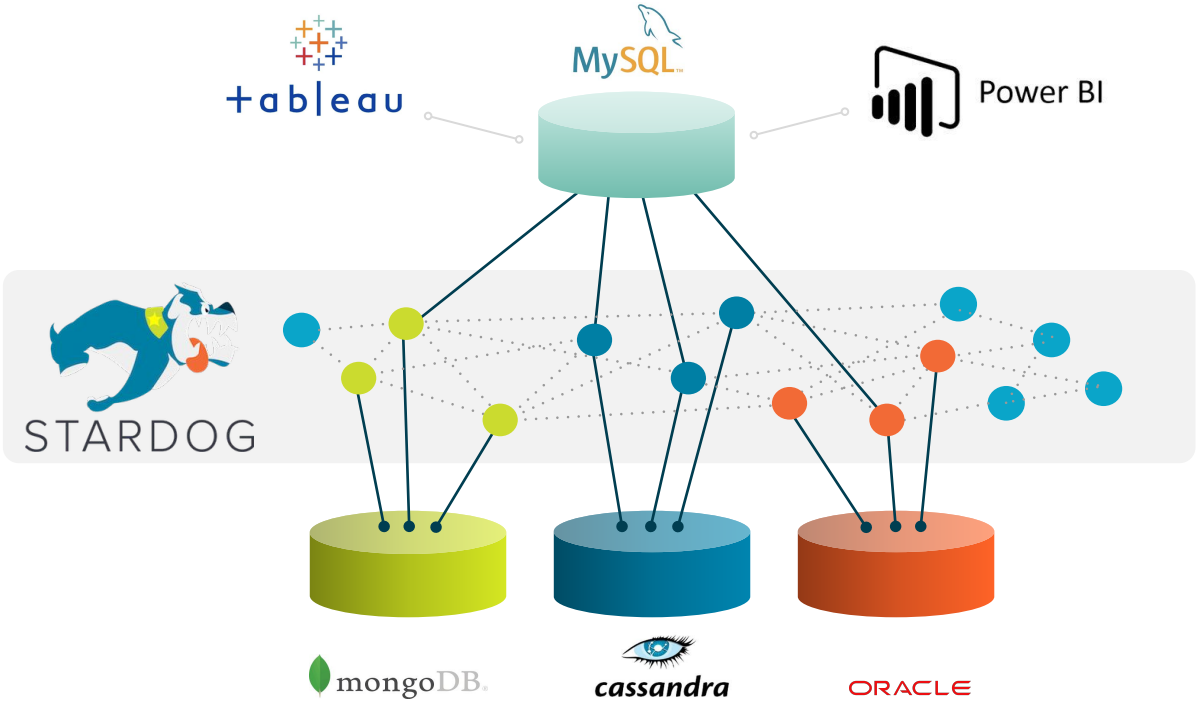
Knowledge Graph

A knowledge graph leverages graph technology and a declarative model to connect, query, and retrieve data.



Examples of data

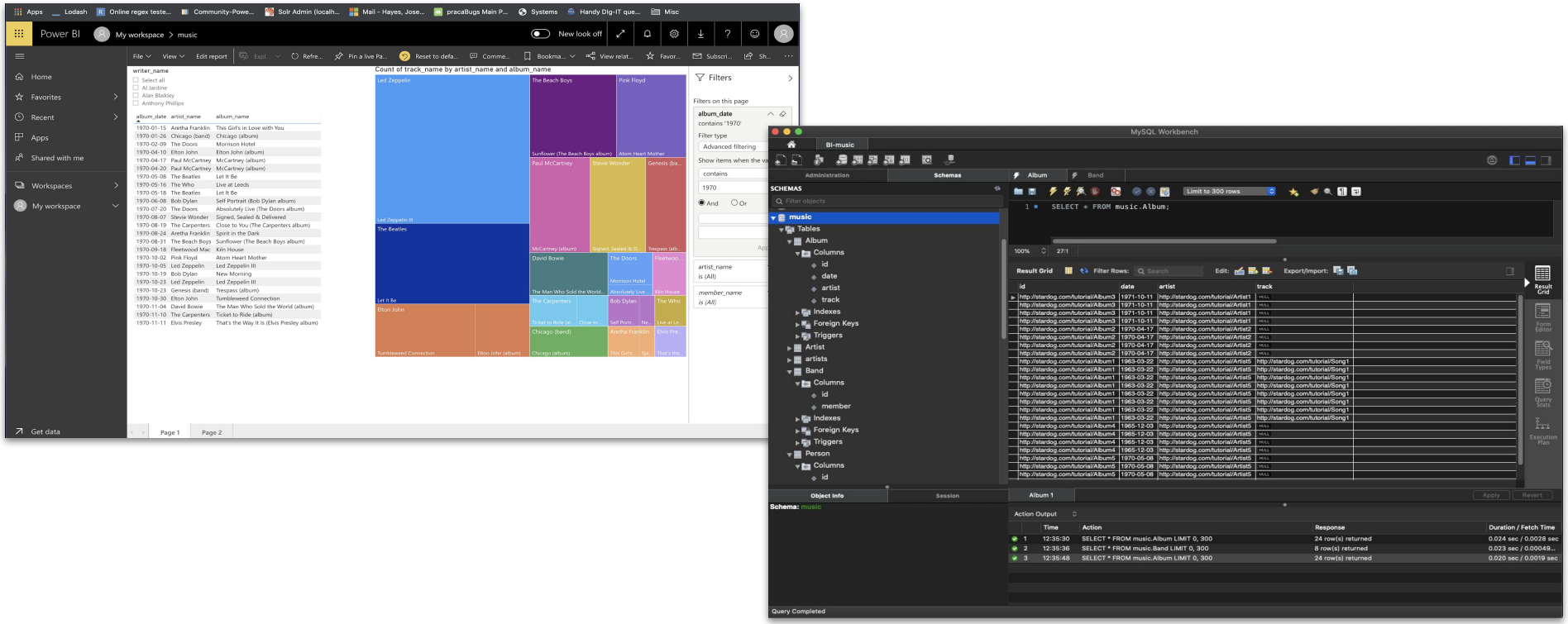
Stardog-SQL Interface & Outline



Stardog SQL Interface & Overview

- Motivation
  - Exploit unified data by established business intelligence tools
  - Expose a JDBC endpoint (MySQL) to a vast number of SQL clients
  - Connect to analytics tools like Apache Spark or RapidMiner

Stardog SQL Interface & Overview



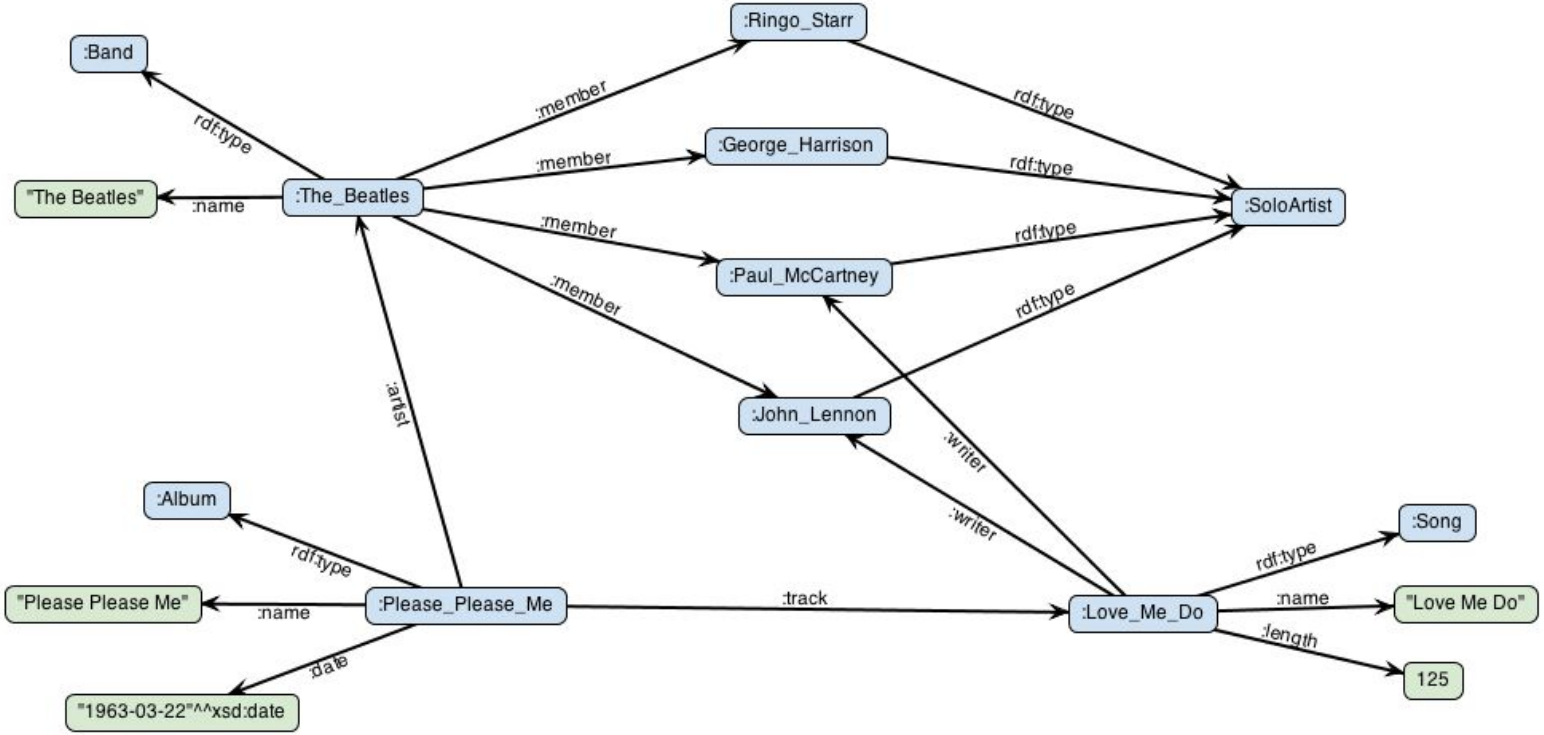
How do Relational and Graph Work Together?

- Many data sources start in relational as an efficient way for storing data of a specific structure with known properties and scale
- Stardog uses knowledge graph to support data unification
  - Maps disparate relational schemas to common graph schema
  - Links related concepts
- Once the data is unified, it's again possible to view it as relations
  - Especially if there are existing reporting tools or dashboards which work with tables

Mappings

STARDOG ACADEMY

Music Data Model



Relational Schema

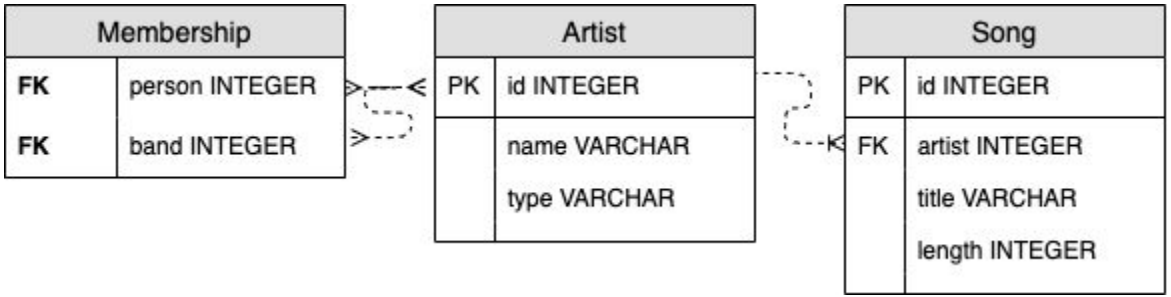


Table Mappings

- Now we need mappings the other way: from RDF to tables
  - Choice 1: create them manually

```
ArtistTableMapping a sql:TableMapping ;
    sql:tableName "Artist" ;
    sql:class :Artist ;
    sql:hasField [
        sql:property :name ;
        sql:fieldName "name" ;
        sql:type xsd:string
    ] .
```

Configuration

- Global settings in `stardog.properties`
  - **`sql.server.enabled=true`**
  - `sql.server.port=5806`
  - `sql.server.commit.invalidates.schema=true`

`true`: changes to the schema mappings visible upon reconnect (default)

`false`: changes effective upon setting DB offline

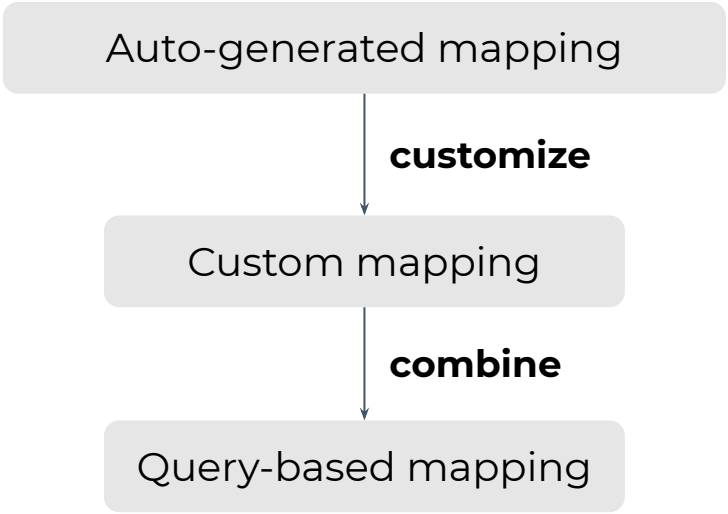
- Further database-level options
  - For example `sql.schema.graph, sql.schema.auto`

Sample Configuration

```
sql.server.enabled=true
sql.server.port=5806
# changes to the schema mappings are visible to new BI connections (true)
# additional DB off/onlining required to re-generate the schema (false)
sql.server.commit.invalidates.schema=true
# named graph to hold the generated/edited mappings
# default: tag:stardog:api:sql:schema
sql.schema.graph=urn:bi:mappings
# automatically generate and use schema mapping (when not manually modified), default
sql.schema.auto=true
```

SQL Mapping Approaches

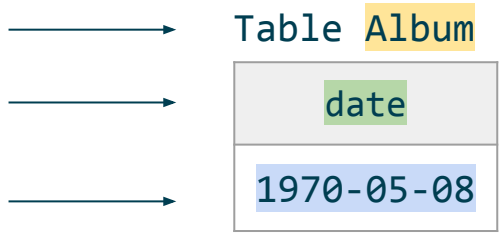
- Reverse mapping from Stardog's graph model to a relational schema



Auto-Generated Mapping

- Schema mapping automatically created when:
  - `sql.schema.auto=true`
  - No custom mappings supplied
- Ontology source (predicates *defined* on a class)
  - `sql.schema.auto.source=owl`

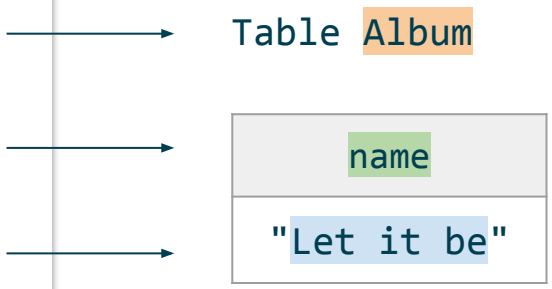
```
:Album a owl:Class . # or rdfs:Class
:date a rdf:type owl:DataProperty ;
    rdfs:domain :Album ;
    rdfs:range  xsd:date .
```



Auto-Generated Mapping II

- Constraints source
  - `sql.schema.auto.source=shacl`

```
:AlbumNameShape
    a sh:NodeShape ;
    sh:targetClass :Album ;
    sh:property [
        sh:path :name ;
        sh:minCount 1 ;
        sh:datatype xsd:string
    ] ;
```



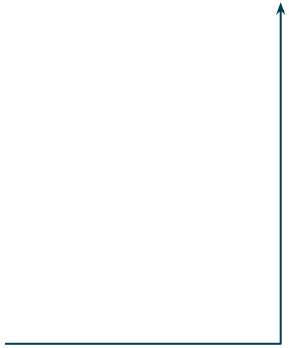
Custom Mappings

- Custom mappings deployed to a dedicated named graph
  - `sql.schema.graph=tag:stardog:api:sql:schema`
  - Same database as used for querying
  - When present auto-mapping is disabled
- Export and augment auto-generated mappings
  - `stardog data model --input owl --output SQL music > mappings.ttl`
- Update custom mappings
  - `stardog query execute music "clear graph <tag:stardog:api:sql:schema>"`
  - `stardog data add -g tag:stardog:api:sql:schema music mappings.ttl`

Sample Mapping

```
:SongTableMapping a sql:TableMapping ;
    sql:tableName "Song" ;
    sql:class :Song ;
    sql:hasField [
        sql:property :length ;
        sql:fieldName "length" ;
        sql:type xsd:integer
    ] ,
    [
        sql:property :writer ;
        sql:fieldName "writer" ;
        sql:refersTo :SongwriterTableMapping
    ] .
```

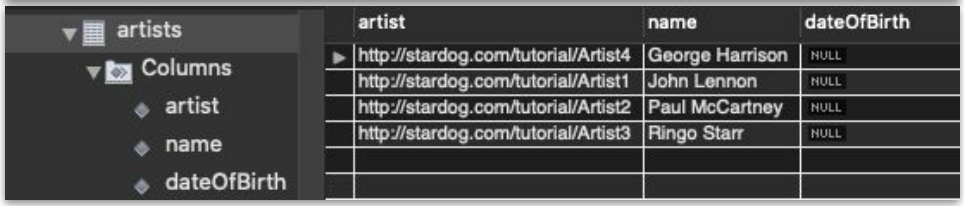
```
:SongwriterTableMapping
    a sql:TableMapping;
    sql:tableName "Songwriter" ;
    sql:class :Songwriter .
```



Query-Based Mapping

- Stored SELECT queries are mapped to SQL tables
- Columns correspond to result variables

```
SELECT ?artist ?name ?dateOfBirth WHERE {
    GRAPH <virtual://music> {
        ?artist
            a :SoloArtist ;
            :name ?name
    }
    OPTIONAL{ ?artist :dateOfBirth ?dateOfBirth }
}ORDER BY ?name
```



Query-Based Mapping II

- Enable reasoning for a stored query
  - Default schema: `set @@reasoning_schema = 'default'`
  - Custom schema: `set @@reasoning_schema = 'other_schema'`

Demo

Learning Objectives

Learning Objectives

- Understand the Stardog BI/SQL server feature
- Use this feature through various methods
- Implement the concepts of the SQL Mapping Syntax

Thank you