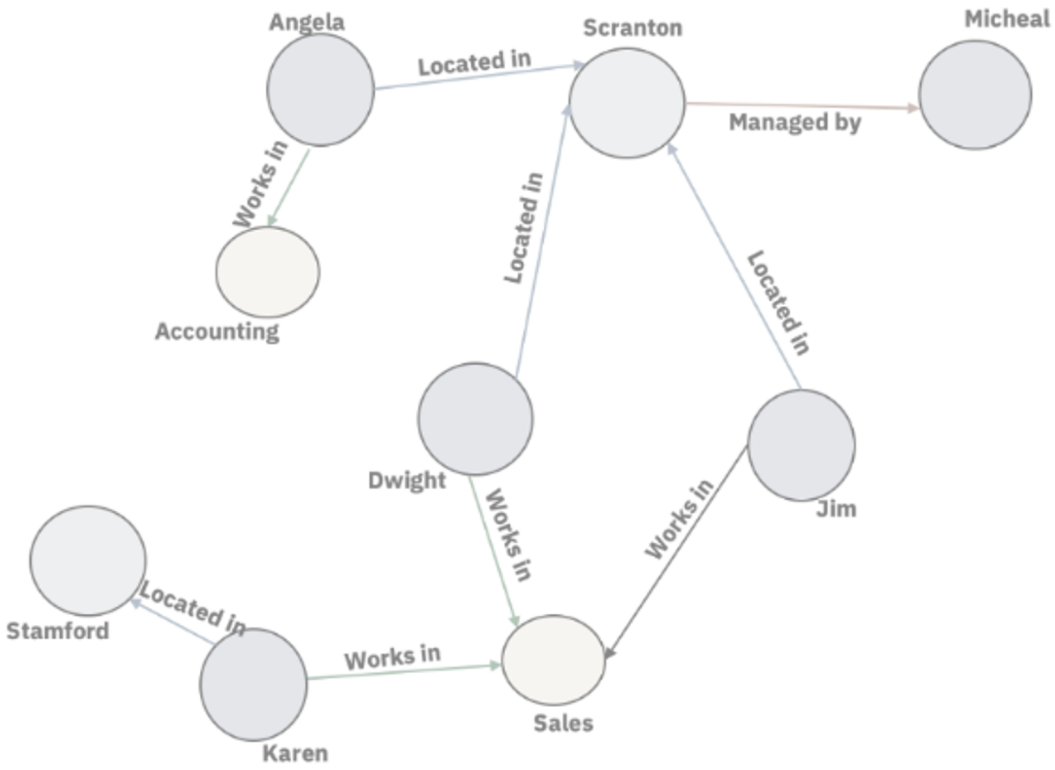
October 8, 2020

About Db2 Graph



The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Agenda

Today we'll introduce Graph databases and touch on a new feature of IBM Db2 11.5.4, the technical preview of **Db2 Graph**.

01

About graphs

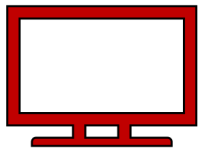
02

Db2 Graph

03

Demo

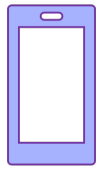
Graph databases are all around us



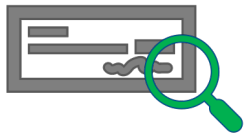
- Your favorite streaming service uses Graph databases to manage all their assets, to understand their users' viewing habits and to create personalized recommendations for movies and shows

- Companies manage the entire supply chain, inventory, orders and user history with Graph databases

- Your airline or hotel booking system may be using Graph databases to manage and recommend new options for users and build reports

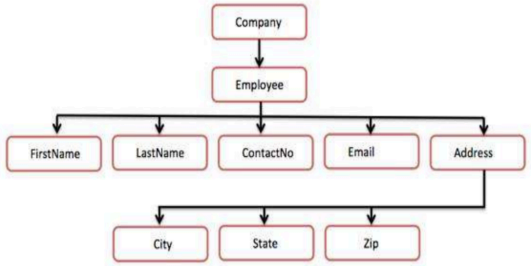


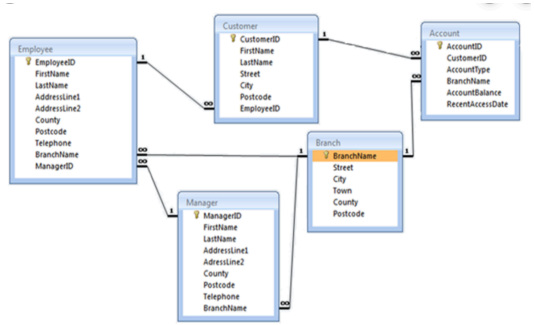
- Your network provider may be using Graph databases to model networks to manage issues, like if a cell tower goes down or in case of a breach

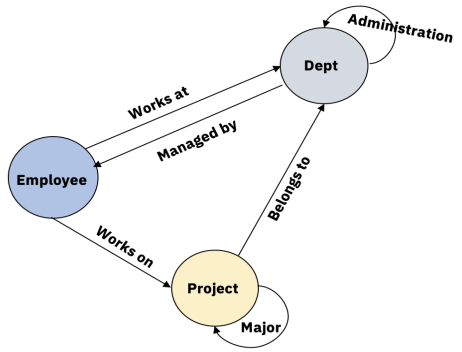


- Your search engine or navigation system was built using Graph principles

How is data stored?







- **NoSQL** (document stores)
  - Stores sets of disconnected data, may be duplication, no ACID compliance.
  - Fast updates and retrievals of sets of data.
- **RDBMS**
  - Relationships defined by joins.
  - Good for transactions, aggregations, transformations.
  - Can’t handle indirect/complex relationships.
- **Graph**
  - Traverses the network to find indirect relationships and patterns.
  - Intuitive to query and visualize from different starting points.
  - Can’t handle fast updates or aggregations.

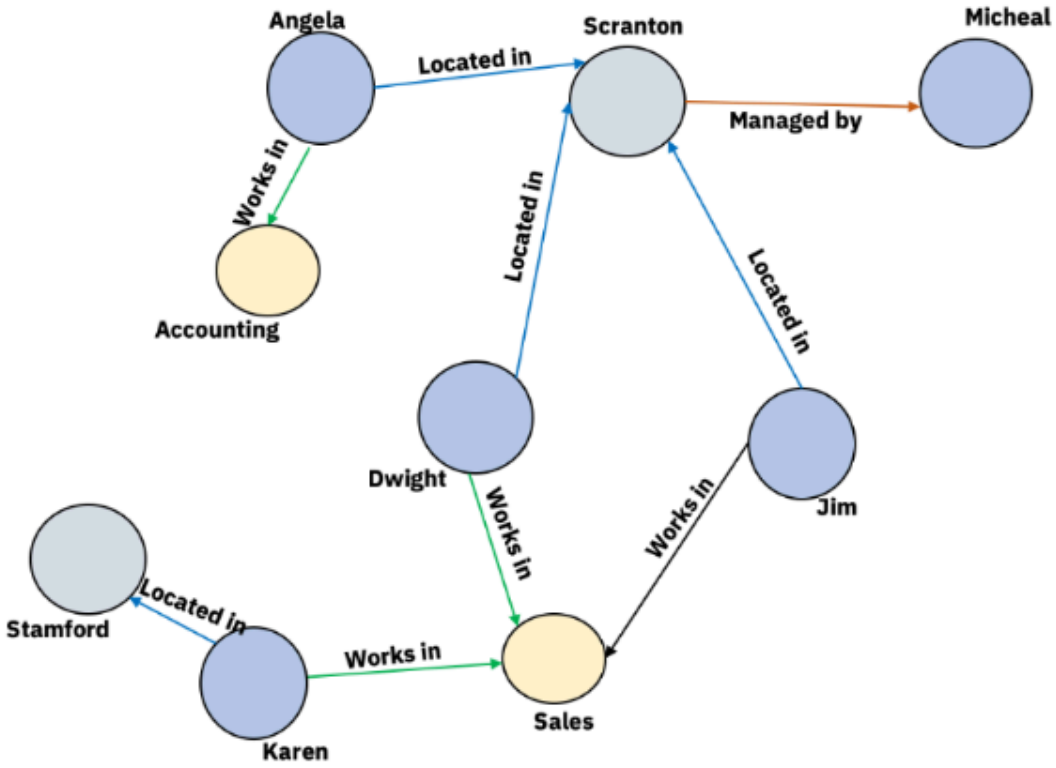
LESS

How connected is your data?

MORE

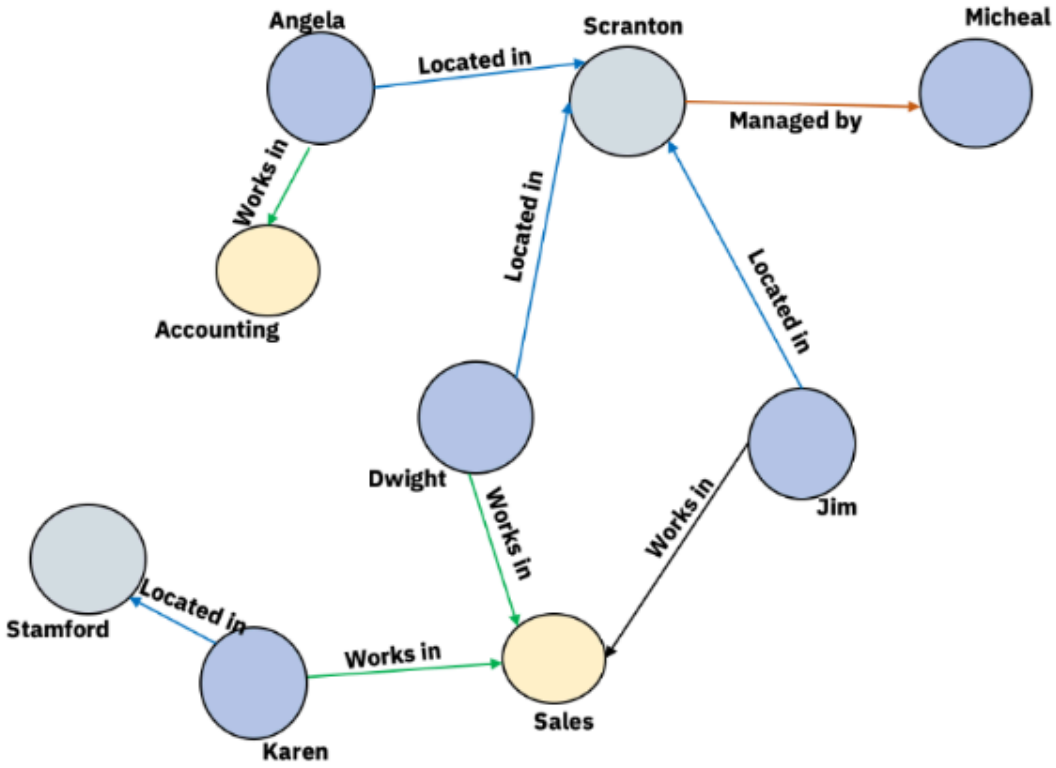
What is a property graph?

- Based on objects called **vertices** and their relationships, called **edges**.
- Vertices have **inbound** and **outbound** edges.
- Edges come **from** one vertex **to** another.
- The collection of vertices and edges define the graph.



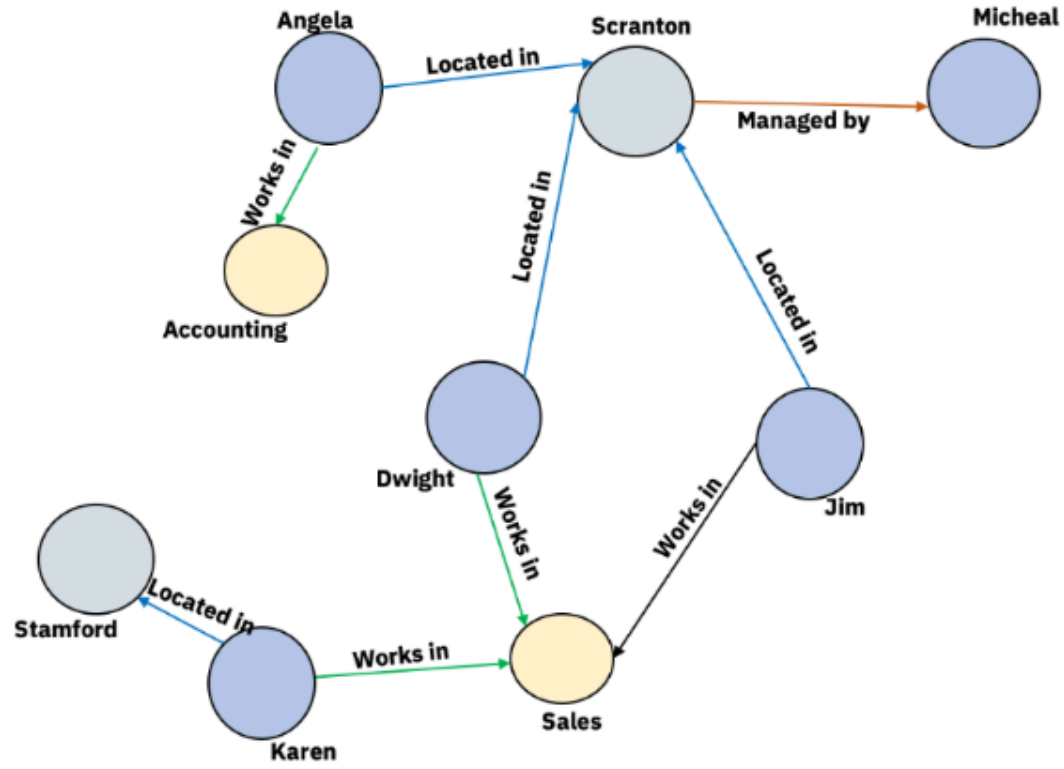
What is a property graph?

- Each vertex or edge has:
  - An **ID** that is unique across the graph
  - A **label** that represents the type of object
  - A set of **properties** that provide additional attributes.



How do you query a graph?

- Queries look at the relationships between vertices.
- An algorithm, the **graph traversal**, starts from a particular set of vertices and ends some defined depth away.
- Filters match ids, labels or properties and constrain the results.



Native vs Non-native graph databases

NATIVE:

- Specialized data store and query engine
- Implements the graph model down to the storage level

EXAMPLES:

- Neo4j, OrientDB, TigerGraph

NON-NATIVE:

- Use existing data store or materializes Graph in-memory
- Uses a complex schema to support both uses

EXAMPLES:

- MSSQL server graph extension, ArangoDB, JanusGraph

What delays adoption of existing solutions?

- **Native graph solutions** that are suitable for graph-only workloads require you to migrate data from existing systems.
- **Hybrid solutions** that duplicate data in-database or in-memory or represent data in complex formats that aren't easily accessible.

- High setup and maintenance costs.
- Relational database are used to solve many graph use cases.
- Duplicate data in another database or in-memory.
  - Inconsistent copies of data.
  - Export/import overhead to load data.
  - Cost of additional storage.

What is Db2 Graph?

- An enterprise grade in-database Graph solution
  - No movement of data into secondary storage to perform graph queries
- No additional cost to existing Db2 licenses
- Queries data and relationships with existing relational tables, supporting Graph analytics and traditional SQL on the same data
- Fetches data in real-time through JDBC

What is Db2 Graph?

- Based on Apache **TinkerPop**
  - TinkerPop is an open-source graph framework
  - **Gremlin** is the graph query language of TinkerPop
  - Db2 Graph is a provider plugin for TinkerPop
- Currently ships in a container that includes the Gremlin Server and Gremlin Console
- Released as technical preview in Db2 11.5.4

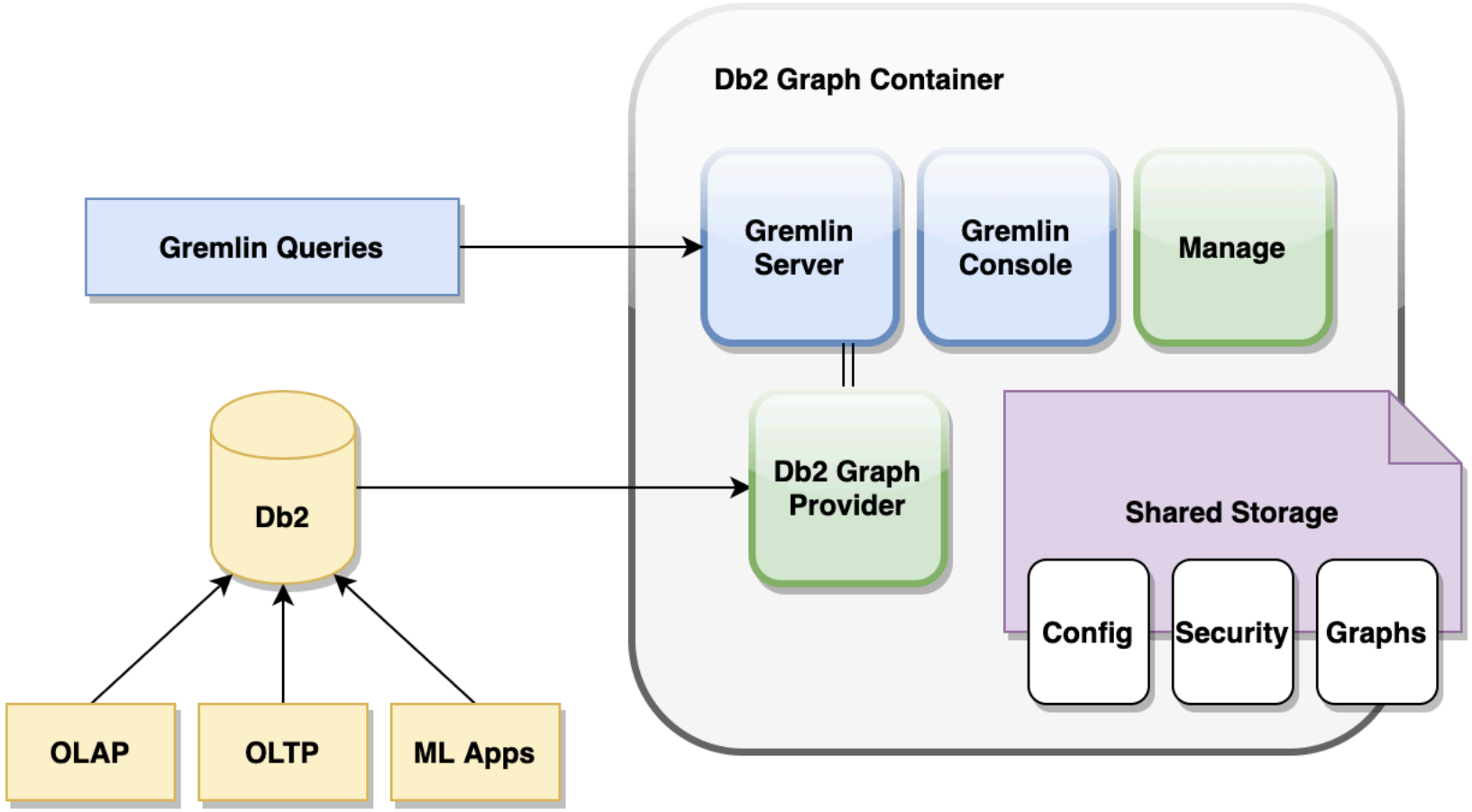
How does Graph differ from SQL?

Essential Operations on Data	RDBMS	GRAPH
Simple relationships between records	●	
Transaction information	●	
Summation and max queries	●	
Business intelligence (aggregations)	●	
Complex queries on interrelated data		●
Deep traversals		●
Indirect relationships (friend of a friend) k-hop traversals		●
Expressive query language and visual results		●
Versioning or auditing of data	●	
Fast execution of complex pattern matching queries		●
Represent multiple versions of the same graph		Db2 Graph

How do you query a graph?

- g.V().hasLabel('DEMO.CLAIM').has('CLAIM_ID', 'C4377').out('DEMO.INSURED_OF_CLAIM').out('DEMO.HAS_DISEASE').path().by(__.valueMap(True))

Db2 Graph Components



How does Db2 Graph work?

- Defines a virtual graph model, the **topology**, on top of Db2 tables
  - Uses referential information to automatically create a graph schema
  - Maps tables or views to graph vertices and edges
- Users create a **topology** using the container's manage command
- Db2 Graph uses the topology to convert Gremlin queries to SQL

How does it work?

- Db2 remains untouched. No change in data, structure or performance.
- Existing applications are uninterrupted.
- Data is fetched by Db2 Graph at time of execution. Data consistency and updates are reflected in real-time.

Mapping relational data to a graph model

- Persists the identification of vertices and edges in a json model.
  - Vertices represent tables with a primary key
  - Edges represent relationships between tables identified by a foreign key
- Views are supported, but not included in the auto-generation and must be manually added.

Patient Table

patientID	name	address	subscriptionID
1	Alice	…	115
…			…

HasDisease Table

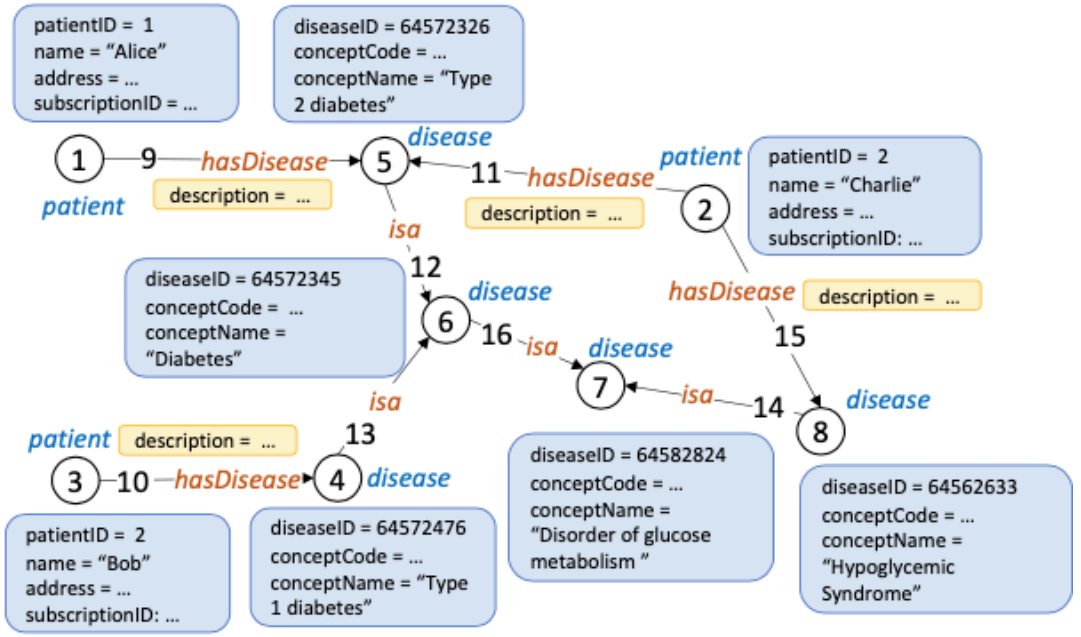
patientID	diseaseID	description
1	64572326	…
…	…	…

Disease Table

diseaseID	conceptCode	conceptName
64572326	44054006	"Type 2 diabetes"
…	…	

DiseaseOntology Table

sourceID	targetID	type
64572326	73211009	"isa"
…	…	



DEMO

Getting started

- To access Db2 Graph visit:
  - https://supportcontent.ibm.com/support/pages/technology-preview-ibm-db2-graph