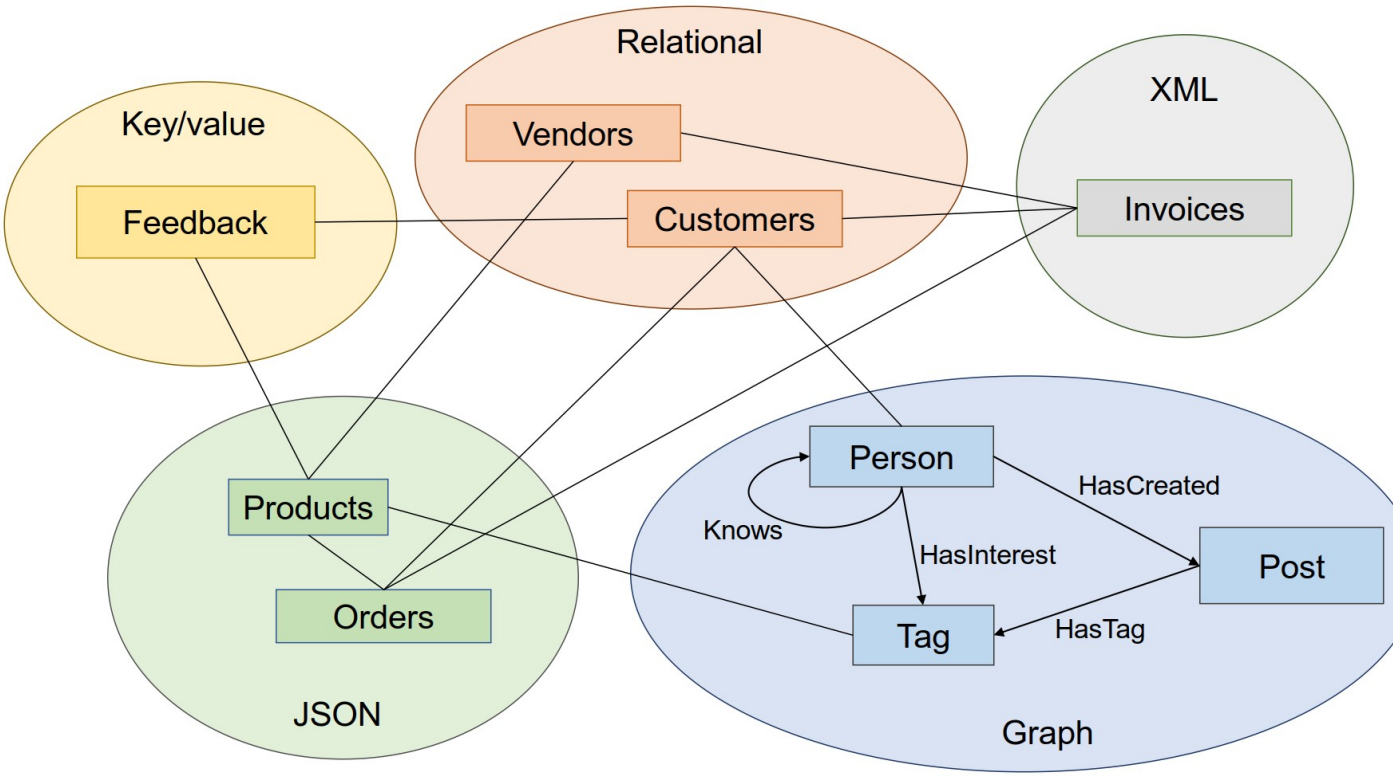
UniBench: Towards Benchmarking Multi-Model DBMS

Motivation

As more businesses realized that data, in all forms and sizes, is critical to making the best possible decisions, we see the continued growth of systems that support massive volume of relational or non-relational forms of data. Unlike traditional database management systems which are organized around a single data model that determines how data can be organized, stored, and manipulated, a multi-model database is designed to support multiple data models against a single, integrated backend. For example, document, graph, relational, and key-value models are examples of data models that may be supported by a multi-model database. Having a single data platform for managing both well-structured data and NoSQL data is beneficial to users; this approach reduces significantly integration, migration, development, maintenance, and operational issues.



A high level overview of the data model. It includes relational, XML, graph, JSON and key-value data models.

Publications

- "UniBench: A Benchmark for Multi-Model Database Management Systems" Accepted in TPCTC 2018.
- "Parameter Curation and Data generation for Benchmarking Multi-model Queries" Accepted in VLDB 2018 PhD workshop.
- "Towards Benchming Multi-model databases" Published in CIDR 2017.

Multi-model queries

- Query 1. For a given customer, find his/her all related data including profile, orders, invoices, feedback, comments, and posts in the last month, return the

category in which he/she has bought the largest number of products, and return the tag which he/she has engaged the greatest times in the posts.

- Query 2. For a given product during a given period, find the people who commented or posted on it, and had bought it.

- Query 3. For a given product during a given period, find people who have undertaken activities related to it, e.g., posts, comments, and review, and return sentences from these texts that contain negative sentiments.

- Query 4. Find the top-2 persons who spend the highest amount of money in orders. Then for each person, traverse her knows-graph with 3-hop to find the friends, and finally return the common friends of these two persons.

- Query 5. Given a start customer and a product category, find persons who are this customer's friends within 3-hop friendships in Knows graph, besides, they have bought products in the given category. Finally, return feedback with the 5-rating review of those bought products.

- Query 6. Given customer 1 and customer 2, find persons in the shortest path between them in the subgraph, and return the TOP 3 best sellers from all these persons' purchases.

- Query 7. For the products of a given vendor with declining sales compare to the former quarter, analyze the reviews for these items to see if there are any negative sentiments.

- Query 8. For all the products of a given category during a given year, compute its total sales amount, and measure its popularity in the social media.

- Query 9. Find top-3 companies who have the largest amount of sales at one country, for each company, compare the number of the male and female customers, and return the most recent posts of them.

- Query 10. Find the top-10 most active persons by aggregating the posts during the last year, then calculate their RFM (Recency, Frequency, Monetary) value in the same period, and return their recent reviews and tags of interest.

Multi-model transactions

- New Order Transaction. (i) create and insert the order, (ii) update the quantity of involved products, (iii) insert the invoice. Hence, this transaction involve three entities: order, product, and invoice.
- Payment Transaction. (i) retrieve the unpaid order, (ii) update the balance of the seller and buyer, (iii) update the order status to paid, (iv) update the related invoice. This transaction involve three entities: order, customer, and invoice.

Download ready-made multi-model dataset

See the detailed schema of Unibench model