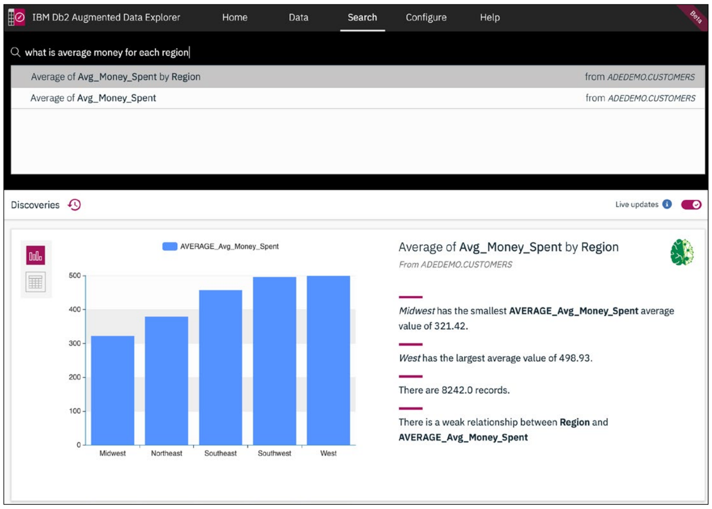
IBM Db2 Augmented Data Explorer

IBM Db2 Augmented Data Explorer

Searching for data and running ad hoc analyses on it can be both difficult and time consuming. First, a data scientist or analyst must identify which database tables contain relevant data. Then they need to construct a SQL query that retrieves the data, which might involve joins and in-database aggregations and calculations. When the data is ready, the data scientist must analyze it and summarize the results.

IBM® Db2® Augmented Data Explorer compresses these steps into a search experience. Any user—not just data scientists and analysts—can search for data and retrieve insights from Db2. It processes natural language requests and returns real-time query results while a user enters a query. These results are also augmented with statistical insights that highlight what is important in the returned data. Db2 Augmented Data Explorer sits somewhere between a business intelligence tool and a data science tool.



what is average money for each region

Average of Avg_Money_Spent by Region — from *ADEDEMO.CUSTOMERS*

Average of Avg_Money_Spent — from *ADEDEMO.CUSTOMERS*

Average of **Avg_Money_Spent** by **Region**

From ADEDEMO.CUSTOMERS

Midwest has the smallest **AVERAGE_Avg_Money_Spent** average value of 321.42.

West has the largest average value of 498.93.

There are 8242.0 records.

There is a weak relationship between **Region** and **AVERAGE_Avg_Money_Spent**

Figure 1: Sample query

In addition, Db2 Augmented Data Explorer can be especially useful for a data scientist who might need to analyze unfamiliar data. Even if they know which columns or fields are in the data, they might not know which are most relevant and they won’t know the distribution of data in those fields or the relationship between them.

Many of the architectural components that make these benefits possible (noted in Figure 2) are discussed in greater detail later in this paper.

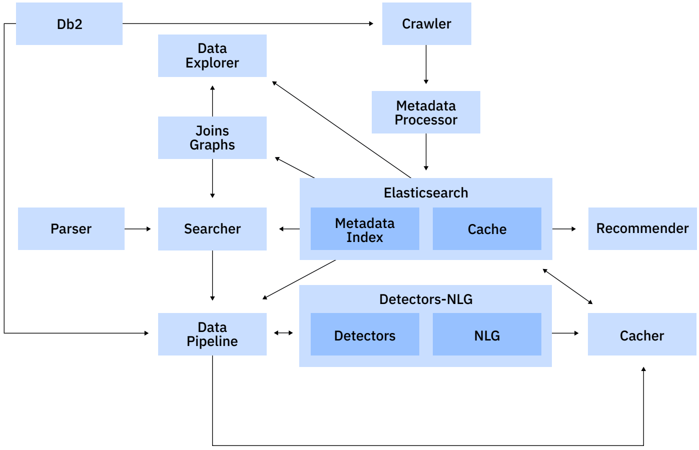


Figure 2: Architecture of Db2 Augmented Data Explorer

Metadata index and cache

Database connections can be slow and, even if the connection is fast, the amount of data could delay retrieval. Furthermore, some metadata that is useful for data analysis, such as a measurement level, is not stored directly in the database.

Db2 Augmented Data Explorer uses Elasticsearch as its local storage layer. Elasticsearch provides several benefits including speed. It is optimized for text searching but also supports keyword-based search for fast lookups and record retrieval. All the metadata and cached results are stored in Elasticsearch.

Crawling

When a user crawls a database, Db2 Augmented Data Explorer analyzes a sample of the data to build a profile of metadata for the tables that are stored in the database. It collects and calculates metadata such as:

- Table and column names
- Measurement level. This is considered when searching for appropriate columns in a query and for statistical analysis of the query’s results
- Unique categorical values
- A score indicating the relative usefulness of a column as a grouping column
- A score indicating the relative likelihood of a column as the target of a statistical analysis. In some cases, the “direction” of a statistical analysis is important

These examples are only part of the metadata stored but they demonstrate its breadth. Calculating this metadata early during crawling is important so that it can be accessed while making real-time query recommendations.

Caching

After metadata crawling, Db2 Augmented Data Explorer initializes caching. Caching entails the storage of aggregated data and the statistical results derived from that data. It significantly improves response time, especially with slower connections and larger databases. There are two types of caching:

- **Pre-emptive caching.** This happens automatically. Using a variety of heuristics, Db2 Augmented Data Explorer generates different queries, runs them through the data pipeline and detectors, and then stores the results in Elasticsearch.
- **Lazy caching.** This happens when a user requests a specific query. It’s similar to browser caching. Results are stored the same way as pre-emptive caching.

Parser

Although a natural option for parsing a user’s question would be an NLP model, there are limitations, such as the need for training data discussed previously. Data could be automatically generated, but these generated questions look like templates. Instead of generating results and feeding them into training, IBM considered using the structure of these templates more directly by creating a grammar.

A context-free grammar was used to parse questions in real time. Training data wasn’t needed for a grammar and the grammar didn’t need to be modified for different data sets. Partial questions can be parsed without the full context to understand the user’s current intent. The result is an abstract representation of an incomplete query that corresponds to the user’s question.

A grammar is not without limitations, though. It is less like natural language and more like a loose syntax. However, the structure and syntax are easy to learn and mimics how many users might ask for queries.

Other language generation options, such as a recurrent neural network for predicting the next word typed, were considered. But as with the NLP solution, would need training data. Investigation into these possibilities are ongoing.

Searcher

Although the solution might have understood the user’s question, the question might not have a valid answer in the current database. Determining if a user’s query can be completed with existing data is essential.

As a user types, Db2 Augmented Data Explorer converts the user’s text into a query. It then reviews metadata stored in Elasticsearch during crawling matching columns in the user’s query with columns in the database. Therefore, suggestions presented to the user will work effectively with the current data.

The best columns from a table are matched for the context in which the column appears in the query. For example, Db2 Augmented Explorer won’t suggest a flag column, a binary categorical field, as a column to aggregate with sum, even if the flag column was stored as numerical values.

It can also search for synonyms and concepts related to the text in the user’s query. For example, if a user types “place”, it will also look for columns that match “region” and “sector.” The same can be done for broader concepts. If a user searches for “demographics,” it will match “gender,” “age” and other similar columns.

Incomplete queries can be augmented with relevant matches as well. If a user types “by” without a following column name, Db2 Augmented Explorer will use metadata to pick out the best grouping columns for the tables that match the other columns in the query.

Suggestions aren’t limited to columns that appear in the same table. After crawling, Db2 Augmented Data Explorer builds a directed graph of the primary and foreign key relationships among the tables. It doesn’t require these relationships to be defined in the database. Referring to the graph, Db2 Augmented Data Explorer can suggest queries across multiple tables. Users don’t need to know that the data is joined or how it was joined.

Because searching is not simple process of text matching, metadata is constantly informing the suggestions that Db2 Augmented Explorer makes.

Data pipeline

Although the suggestions that are presented to the user are complete and valid queries for the database, the data from the original query isn’t sufficient for deeper analysis. Db2 Augmented Data Explorer uses a library of analytic detectors to generate statistical results and interpret them.

Some detectors require more data than would be returned from the original query. One detector may run an analysis of variance (ANOVA) when a user requests an aggregation of a continuous column, such as the mean of sales, by a categorical column. The ANOVA analysis requires more than the mean, and also needs the variance and the counts. Therefore, before Db2 Augmented Data Explorer runs the query against the database, it checks if the detectors need supplemental aggregations. It then augments the original query and runs these aggregations in-database.

The data pipeline also augments the data after it’s returned from the database and detectors. If the query involves a date and a continuous column, Db2 Augmented Data Explorer runs time series analysis on the data. It can also add forecasting results. This continual processing and augmentation of the data is handled by the data pipeline.

Detectors and natural language generation (NLG)

Analytic Detectors
The detectors library processes data that’s passed to it and generates a set of “facts” that describes the data. A client of the detectors library can pass in arbitrarily shaped data, and the detectors library will figure out which analyses are appropriate for the shape and type of data. So, in some cases, the detectors might run customized chi-square tests to determine whether two categorical fields are related to each other. If they are, the detectors run additional tests to find which categorical combinations are driving the relationship.

The detectors are optimized for speed, so techniques that require only milliseconds to run are favored. They are based on in-house algorithms and developed with a combination of custom code and open- source tools like Statsmodels and SciPy. Both statistical significance and effect size are considered to ensure that the results are meaningful.

Further, detectors also generate basic summaries of the data. The full collection of facts is akin to a collection of facts about a physical entity such as a country or a person. In the case of the detectors, the collection of facts describes the data.

Natural language generation

However, this description demands some specialized knowledge. To help those who lack this knowledge, a detectors-nlg library was created. The detectors-nlg library generates natural language descriptions of the detector facts. This library reads in a collection of detector facts and outputs a narrative that highlights the key facts.

The detectors-nlg library is template-based, but randomized to reduce monotony and promote storytelling through data. Off-the-shelf solutions lack a focus on analytics so templates were defined around detector facts. Raw facts could then be turned into a summary that all users could understand.

The library interprets detector facts to determine which group of facts are most important. For example, some detector results include a group of facts related to the association of two fields while another group of facts is related to the predictive relationship between the fields. Based on the specific results, the detectors-nlg library determines which group is best for reporting.

Open source libraries are used to handle grammar, punctuation and capitalization to help ensure that the results read well. The detectors-nlg library can also handle special cases since one template doesn't fit all data conditions. For example, if there are many records with the same maximum value, Db2 Augmented Data Explorer will recognize this situation and report only a couple of the records rather than all of them.

Recommender

Chatbots operate on a pull model. A user asks a question and the chatbot responds. This works well for many cases, but if a user is not familiar with the capabilities of a chatbot, they won't know which questions can be asked.

To avoid a similar issue with Db2 Augmented Data Explorer—or a lack of user familiarity with the data—a push model was added to Db2 Augmented Data Explorer. As discussed previously, Db2 Augmented Data Explorer automatically generates queries and caches the analytic results for fast retrieval. This same system can be used to push the results to the user. When results are cached, they are ranked with facts from the detectors, allowing the most useful ones to be recommended.

In the future, analytical ranking will be combined with usage habits that Db2 Augmented Data Explorer has learned to recommend more targeted results.

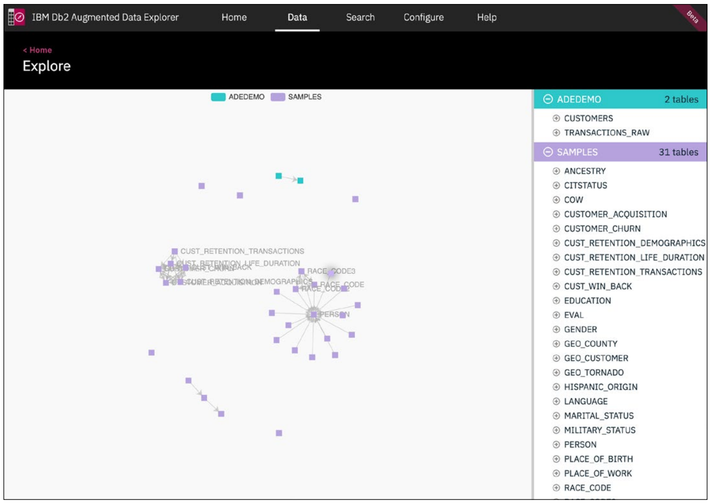


Figure 3: Data explorer

Data explorer

Another idea benefiting users who don't know where to begin is the data explorer, an interactive interface to help obtain an overview of the data and navigate it like a map. Users can zoom in and see details about a table or column and even generate a query from a column. The data explorer was born out of the directed graph, which defines the table relationships. The data explorer shows these relationships and pinpoints which tables are the center of a database join network. This is fast and fluid because Db2 Augmented Data Explorer refers to the data stored in Elasticsearch rather than communicating with the database directly.

Planned and incubating features

Research into Db2 Augmented Data Explorer improvements are ongoing. For example, integration of results into users' workflows, exporting static visualizations and NLG and dynamic code in a Jupyter notebook.

Analytics and augmentation are being investigated as well. Machine learning models were originally eschewed because of the focus on speed, but pre-built models could improve the relevance of insights. Augmenting users' original data with data that is derived internally and externally is also being considered. Moreover, a hybrid model for interpreting queries, one that fuses a grammar-based approach and a traditional NLP approach is being considered, along with Db2 Augmented Data Explorer as the backend of a chatbot..

More information

For more information about Db2 Augmented Data Explorer or to try it for yourself, check out our website.

IBM Corporation
New Orchard Road
Armonk, NY 10504
Produced in the
United States of America
2019

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

All client examples cited or described are presented as illustration of the manner in which some clients have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions. Contact IBM to see what we can do for you.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided. The client is responsible for ensuring compliance with laws and regulations applicable to it. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the client is in compliance with any law or regulation.