IDUG

2022 EMEA Db2 Tech Conference

Now You See It, Unveil New Insights Through SQL Data Insights

Akiko Hoshikawa

IBM

Platform: Db2 for z/OS

Disclaimer

Agenda

Introduction of Db2 13 SQL Data Insights

Technology behind of SQL Data Insights

Understanding Semantic AI queries

Using Semantic AI queries

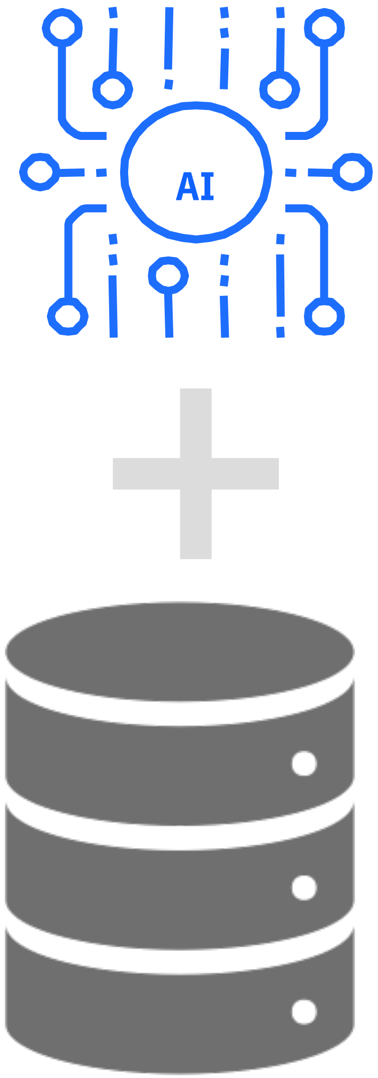
Steps enabling SQL Data Insights

Summary

Q&A

SQL Data Insights

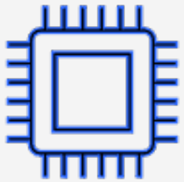
An industry-first relational database with embedded AI capabilities



Infuse NLP directly into your database on existing data to discover hidden information

Minimizes complexity of deploying AI into your applications

Single model used for a range of inferencing task over multiple fields



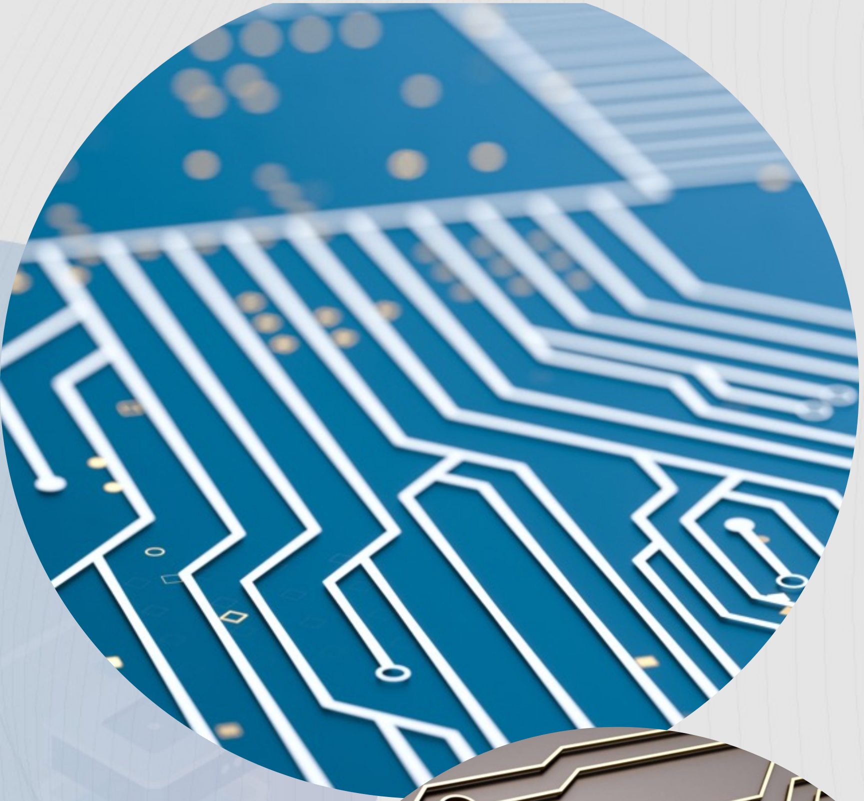
Exploits zIIPs and IBM Z acceleration

Semantic SQL Functions

Initial set of AI Built-In Functions available in Db2 13

Cognitive Intelligence Query	Functional Description	Db2 functions
semantic **similarity and dissimilarities**	• **Matching** rows/entities based on overall meaning (similarity/dissimilarity) • **Suggest** choices for incorrect or missing entities	AI_SIMILARITY
semantic **Clustering**	• **Find** entities/rows based on relationships between attributes in a given set • Example: Find animals similar to (lion, tiger, panther)	AI_SEMANTIC_CLUSTER
Reasoning Analogy	• **Find** entities/rows based on relationships between attributes • Example: Moon : Satellite :: Earth; ?	AI_ANALOGY

Technology Behind of SQL Data Insights



SQL Data Insights: Core Concepts

Unsupervised Neural Network Approach for Natural Language Processing: Word Embedding

- Captures word meaning as collective contributions of words (tokens) in the neighborhood
- Generates semantic representations of words (tokens) using vectors (Vector Embedding)
- Semantic similarities between words (tokens) measured using distance between vectors

Extending Vector Embedding Approach to structured databases: Database Embedding

- Every database column value, irrespective of its column type, converted to a text token
- View a database record as an unordered English-like sentence (bag-of-words) of text tokens
  - Every token is equally related to other tokens in the “sentence”, irrespective of their position
  - Tokens related to unique primary keys and NULL values are treated differently
- Semantic model infers meanings (behavior) of database column values based on their neighboring column values (e.g., within a table row, and across table rows)
- Exploit the trained model to enable new SQL semantic queries that operate on the relational data based on the inferred meaning, not using values

Relationship Hidden in a Table

CustID	Date	Merchant	State	Category	Items	Amount
CustA	9/16	Store-X	NY	Fresh produce	Bananas	80
CustA	9/16	Store-X	NY	Fresh produce	Apples	120
CustD	9/16	Store-Z	NY	Stationary	Crayons	50
CustD	9/16	Store-Z	NY	Stationary	Folders	150
CustC	10/16	Store-X	CT	Fresh produce	Bananas	100
CustC	10/16	Store-X	CT	Fresh produce	Oranges	100

– Which customer’s behavior is more similar to Cust-A’s behavior ?

– What makes you to think so?

Relationship Hidden in a Table

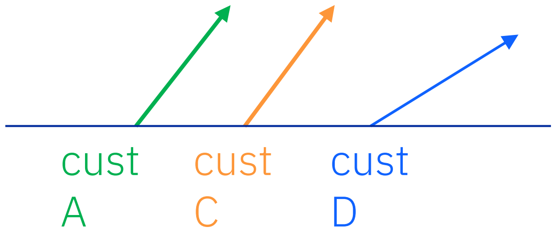
CustID	Date	Merchant	State	Category	Items	Amount
CustA	9/16	Store-X	NY	Fresh produce	Bananas	80
CustA	9/16	Store-X	NY	Fresh produce	Apples	120
CustD	9/16	Store-Z	NY	Stationary	Crayons	50
CustD	9/16	Store-Z	NY	Stationary	Folders	150
CustC	10/16	Store-X	CT	Fresh produce	Bananas	100
CustC	10/16	Store-X	CT	Fresh produce	Oranges	100

Textification : transform values to text token

Txn1 custID_custD Date_9/16 Merchant_ Store-Z State_NY Category_Stationary Items_Folders Amount_1

Generation of "meaning vector" for every column value

custA is similar to custC due to similar purchasing behavior.



- If there is no primary key, row-ID (Txn1 above) will be generated and represent other column values in the same row.
- Meaning vector of the primary key captures the meaning of an entire row.
- Meaning of non-primary key value contributes correctively to its neighbors (e.g. NY is associated with Bananas and Crayons)

Relationship Hidden in a Table

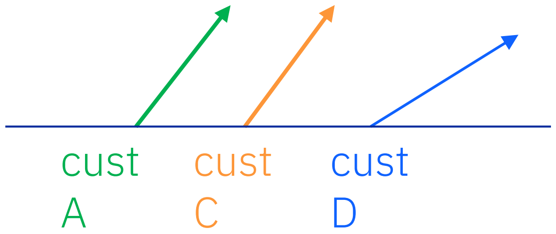
CustID	Date	Merchant	State	Category	Items	Amount
CustA	9/16	Store-X	NY	Fresh produce	Bananas	80
CustA	9/16	Store-X	NY	Fresh produce	Apples	120
CustD	9/16	Store-Z	NY	Stationary	Crayons	50
CustD	9/16	Store-Z	NY	Stationary	Folders	150
CustC	10/16	Store-X	CT	Fresh produce	Bananas	100
CustC	10/16	Store-X	CT	Fresh produce	Oranges	100

Textification : transform values to text token

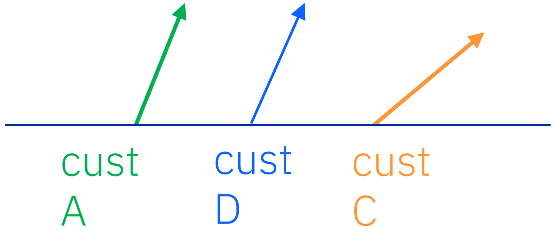
Txn1 custID_custD Date_9/16 Merchant_ Store-Z State_NY Category_Stationary Items_Folders Amount_1

Generation of "meaning vector" for every column value

custA is similar to custC due to similar purchasing behavior.



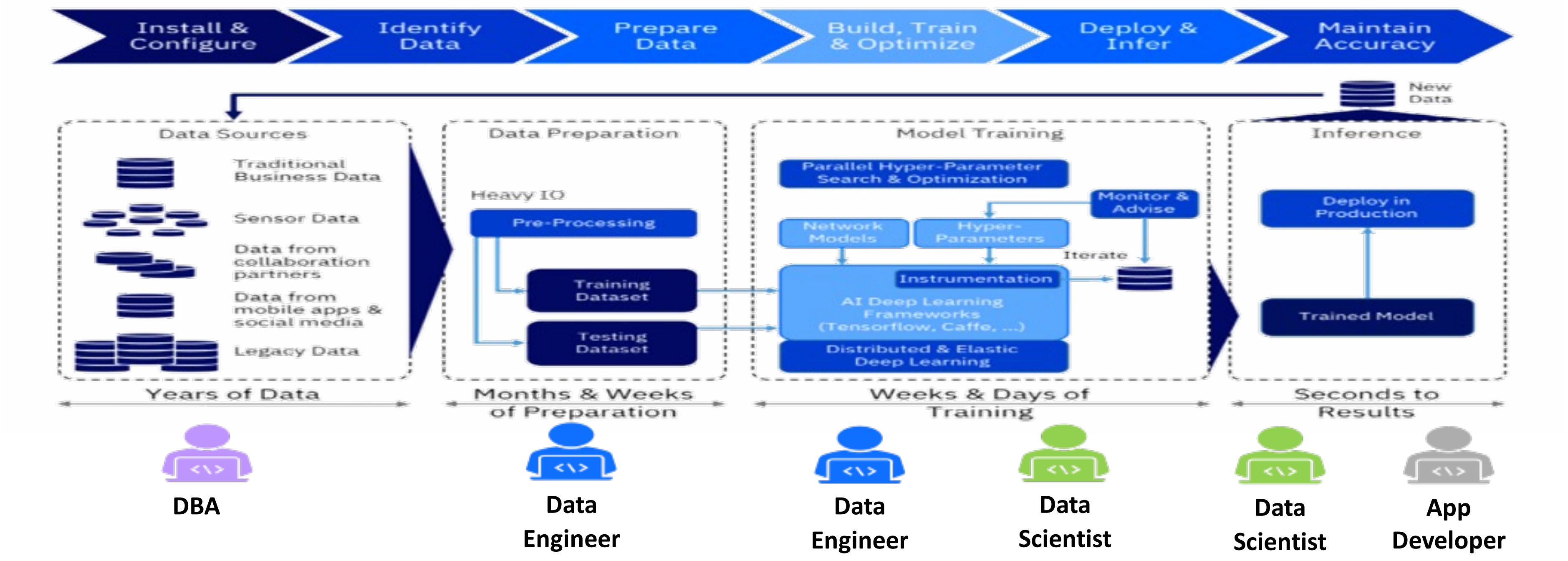
(Withtout Category/Items)
custA is similar to custD due to similar behavior



- If there is no primary key, row-ID (Txn1 above) will be generated and represent other column values in the same row.
- Meaning vector of the primary key captures the meaning of an entire row.
- Meaning of non-primary key value contributes correctively to its neighbors (e.g. NY is associated with Bananas and Crayons)

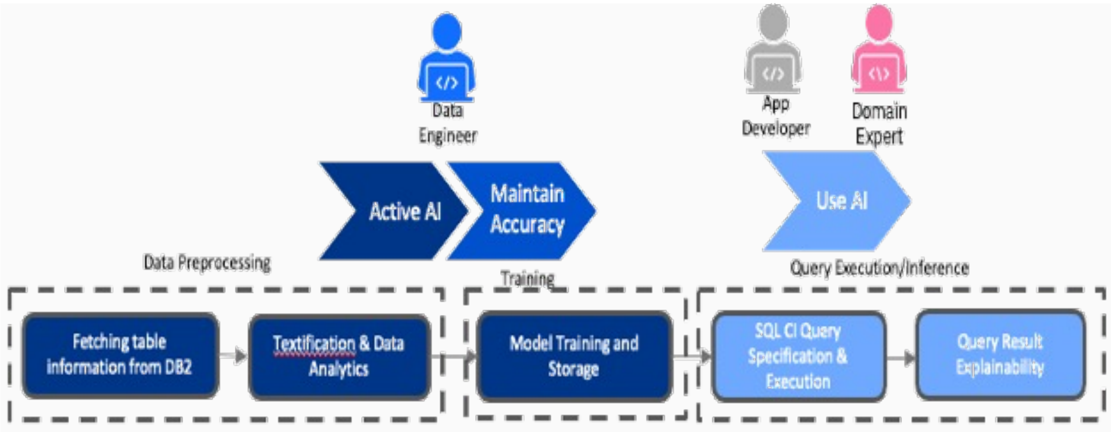
Extract greater value from Db2 for z/OS data

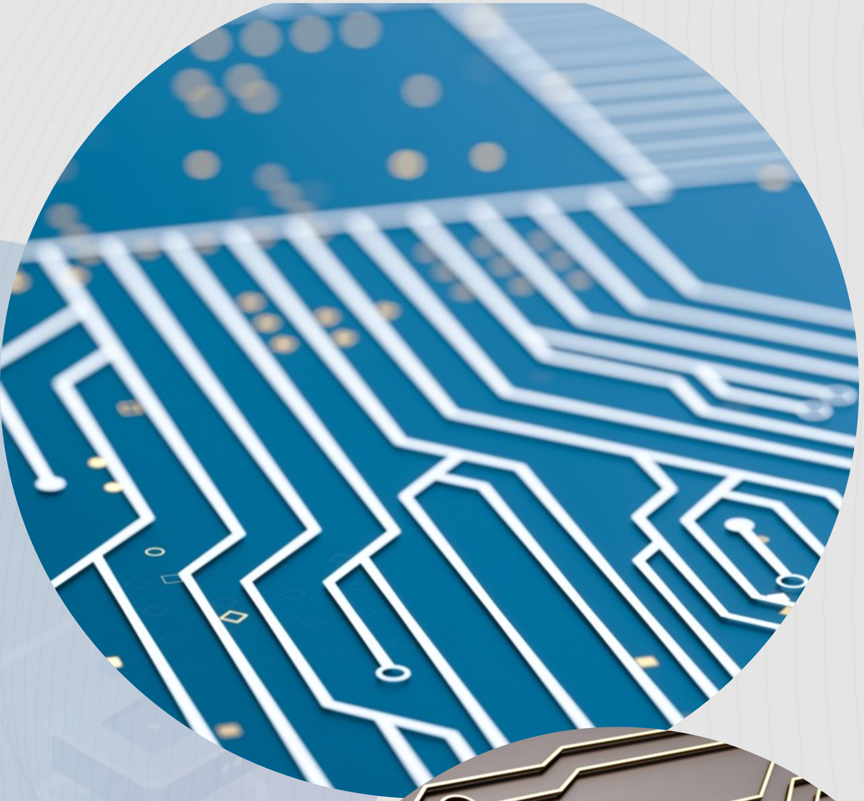
Traditional AI models are complex to build and serve a single narrow purpose



VS.

Build Neural Network powered relationship maps using unsupervised training over (unlabeled) structured data





Semantic AI Functions

AI_SIMILARITY

```
AI_SIMILARITY (expression-1  USING MODEL COLUMN column-name,
               expression-2  USING MODEL COLUMN column-name )
```

AI_SIMILARITY('APPLE', 'RASPBERRY' USING MODEL COLUMN FRUIT)

It computes a similarity score using the values returned by expression-1 and expression-2.
Results of AI_SIMILARITY – floating point number between -1.0 and 1.0
1.0 means very similar or same, -1.0 means very dissimilar

Find top 5 customer IDs that are the most similar to a customer “3668-QPYBJ” who closed his account
note : customerID is defined as a primary key

```
SELECT  AI_SIMILARITY(X.customerID,'3668-QPYBK’ USING MODEL
COLUMN customerID )  AS SimilarityScore, X.*
FROM CHURN X
WHERE X.customerID <> '3668-QPYBK'
ORDER BY SimilarityScore DESC
FETCH FIRST 5 ROWS ONLY;
```

SIMILARITYSCORE	CUSTOMERID	GENDER	SENIORCITIZEN	PARTNER	DEPENDENTS	TENURE	PHONESERVICE	MULTIPLELINES	INTERN
0.902809739112854	2207-OBZNX	Male	0	No	No	7	Yes	No	DSL
0.8648061752319336	2108-XWMPY	Male	0	No	No	3	No	No phone service	DSL
0.8551765084266663	6304-IJFSQ	Male	0	No	No	3	Yes	No	DSL
0.8473891615867615	5493-SDRDQ	Male	0	No	No	2	Yes	No	DSL
0.8069272637367249	7580-UGXNC	Female	1	No	No	2	Yes	No	DSL

AI_SIMILARITY – Dissimilarity Query

Find top 5 customer IDs that are the least similar to a customer “3668-QPYBJ” who closed his account
note : customerID is defined as a primary key

```
SELECT  AI_SIMILARITY(X.customerID,'3668-QPYBK’ USING MODEL
COLUMN customerID )  AS SimilarityScore, X.*
FROM CHURN X
WHERE X.customerID <> '3668-QPYBK'
ORDER BY SimilarityScore ASC
FETCH FIRST 5 ROWS ONLY;
```

SIMILARITYSCORE	CUSTOMERID	GENDER	SENIORCITIZEN	PARTNER	DEPENDENTS	TENURE	PHONESERVICE	MULTIPLELINES	INTERNETSERVICE
-0.19289052486419678	6050-FFXES	Female	0	Yes	No	69	Yes	Yes	Fiber optic
-0.1552256941795349	6766-HFKLA	Female	0	Yes	No	56	Yes	Yes	Fiber optic
-0.1492832899093628	8433-WPJTV	Female	1	Yes	Yes	65	Yes	Yes	Fiber optic
-0.13930177688598633	4128-ETESU	Female	1	Yes	No	47	Yes	Yes	Fiber optic
-0.12915533781051636	1400-WIVLL	Male	0	Yes	No	57	Yes	Yes	Fiber optic

Sponsor User's Test

Find the most similar 5 car manufacturers as Ferrari in the car data base

```
SELECT DISTINCT AI_SIMILARITY(MAKE,'Ferrari') as SCORE, MAKE
FROM CARS
WHERE MAKE <> 'Ferrari'
ORDER BY 1 DESC
FETCH FIRST 5 ROWS ONLY
---------+---------+---------+---------+---------+-
      Score                      MAKE
---------+---------+---------+---------+---------+-
+0.735175132751 4648E+00  Lamborghini
+0.6999126672744751E+00  Rolls-Royce
+0.6649318337440491E+00  Bentley
+0.6472378969192505E+00  Corvette
+0.6257274746894836E+00  McLaren
```

https://www.kaggle.com/datasets/ander289386/cars-germany

Insurance Use Case

Insurance company realizes that they are undercharging a policy holder and want to find customers since 2015 that are similar to him to avoid losses

```
SELECT *
FROM
(SELECT C.*,
AI_SIMILARITY(DRIVERS_LICENSE_NUMBER,
'339 713 155’) AS SIMILARITY
FROM IBM.INSURANCE C)
WHERE
HEATING_LAST_UPDATE_YEAR>’2015’
ORDER BY SIMILARITY
DESC
FETCH FIRST 20 ROWS ONLY
```

IBM Synthetic Data – Insurance Underwriters

AI_SEMANTIC_CLUSTER

```
AI_SEMANTIC_CLUSTER (member-expression  USING MODEL COLUMN column-name,
                                        clustering-expressions)
```

```
AI_SEMANTIC_CLUSTER('STRAWBERRY' USING MODEL COLUMN FRUIT, 'RASPBERRY', 'BLACKBERRY', 'BLUEBERRY')
```

computes a clustering score using the values returned by clustering-expressions
Results of AI_SEMANTIC_CLUSTER – floating point number between -1.0 and 1.0
Higher score means a better clustering of member-expression among the clustering-expressions

Based on a group of customers who have high valued houses and no recent updates, find similar customers to increase premium

```
SELECT  C.*,
AI_SEMANTIC_CLUSTER(C.DRIVERS_LICENSE_NUMBER ,'Q08670943', '543877806', 'T30381936') AS SIMILARITY
FROM AAMININ.INSURANCE C
WHERE C.DRIVERS_LICENSE_NUMBER NOT IN ('Q08670943', '543877806','T30381936')
ORDER BY SIMILARITY DESC
FETCH FIRST 20 ROWS ONLY
```

AI_ANALOGY :

AI_ANALOGY (source-1, target-1, source-2, target-2)

AI_ANALOGY('STRAWBERRY' USING MODEL COLUMN FRUIT, 'RED',
'LEMON', 'YELLOW')

computes an analogy score using the values returned by the arguments. Higher the score, a better analogy than a lower score.
Results of AI_ANALOGY – floating point number, NOT bounded by -1.0 and 1.0

Analyze the relationships between length of contract and internet service subscriptions

```
SELECT DISTINCT
    AI_ANALOGY('Month-to-month' USING MODEL COLUMN CONTRACT,
               'Fiber optic' USING MODEL COLUMN INTERNETSERVICE,
               'Two year',
                INTERNETSERVICE) AS ANALOGY_
        X.INTERNETSERVICE
 FROM CHURN X
 WHERE X.INTERNETSERVICE<>'Fiber optic'
 ORDER BY ANALOGY_SCORE DESC
```

ANALOGY_SCORE	INTERNETSERVICE
0.8413964921922206	DSL
0.6485916530516833	No

Insurance Use Case

Find risky customers in Oklahoma based on a risky customer found in Kansas

```
SELECT * FROM
(SELECT AI_ANALOGY (
‘Kansas’ USING MODEL COLUMN DRIVERS_LICENSE_STATE,
‘Q06-25-5829' USING MODEL COLUMN DRIVERS_LICENSE_NUMBER,
‘Oklahoma’ USING MODEL COLUMN DRIVERS_LICENSE_STATE,
 DRIVERS_LICENSE_NUMBER) AS ANALOGY_SCORE ,C.*
FROM IBM.INSURANCE C)
ORDER BY ANALOGY_SCORE DESC
FETCH FIRST 20 ROWS ONLY ;
```

IBM Synthetic Data – Insurance Underwriters Use case

SQL Data Insights - Potential Use Cases

Finance (Consumer Banking, Investment Advisors)

- Find customers with similar transactions
- Non-performing Asset Identification (NPA)

Fraud detection

- Anti money laundering
- Account take-over

Insurance

- Identify similar/dissimilar claims
- Evaluate risk profiles by analyzing patient profiles (e.g., symptoms, diagnosis…)

IoT

- Find households/hotel rooms with similar energy consumption patterns

Customer analytics

- Find similar customers based on buying patterns
- Customer Churn Analytics

Advanced sales prediction using external data

- Predict sales of new products to existing customer base

IT incident ticket analysis

- Find accounts with similar ticket patterns

HR

- Find employees with similar skills and similar/different experience

Entity resolution/Data imputation for data quality

- Identify multiple instances of a single customer across multiple data sources

Any use case in your business?

Customer Retention Analysis

- Business needs – retention program at telecom company
  - Reduce the customers who leave the service.
- Data stored in databases
  - Customer information, Service subscription, Billing
- Persona – a business analyst
  - Data analysis skill (SQL skill) – good
  - Data science skill – limited
- Scenario
  - Use AI semantic queries to perform analysis.
    - Identify similar customers who might leave the business based on the customer's record who had already left
    - Identify the common pattern among high-risk customers
    - Identify the set of customers who are not likely leaving and understand the pattern