

Skills 101

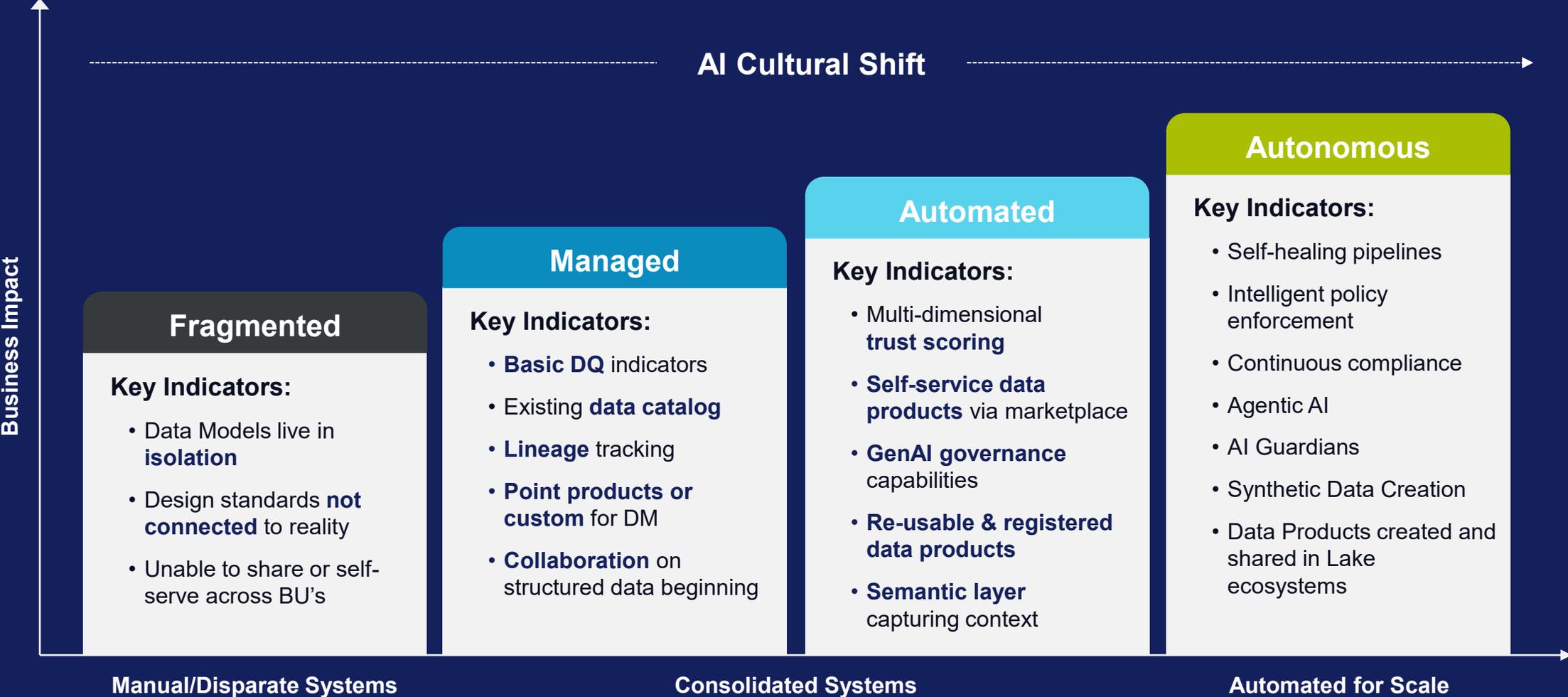
Fix the Data. Then Scale the AI. Governance and Modeling in Action

 Quest

Geoff Hodgkinson
Solutions Engineer
Geoff.Hodgkinson@quest.com



AI Data Readiness Curve



“

**Speed without Trust is a liability.
Trust without speed is a bottleneck.
You need both.**

”

Stephan Covey



AI Capabilities

Data Modeling

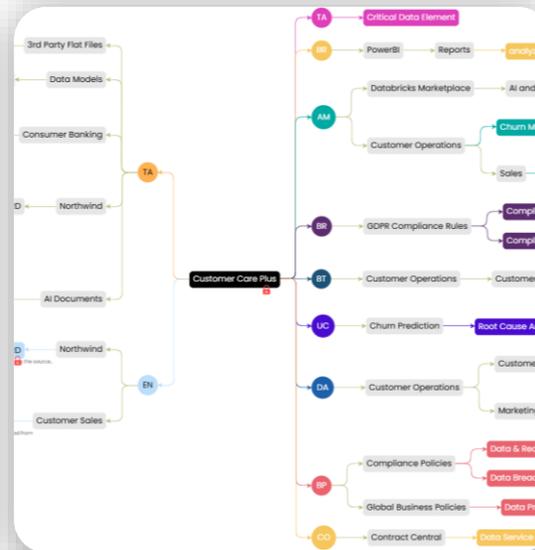
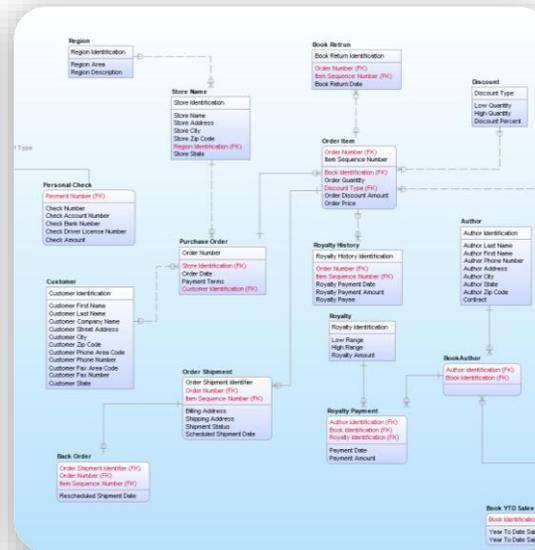
- Generating models
- Reviewing schema

Data Intelligence

- Generating business term definitions
- Discovering data quality issues
- Generating data quality measures

Automatic Data Product Factory (ADPF)

- Generating and enriching Data Products

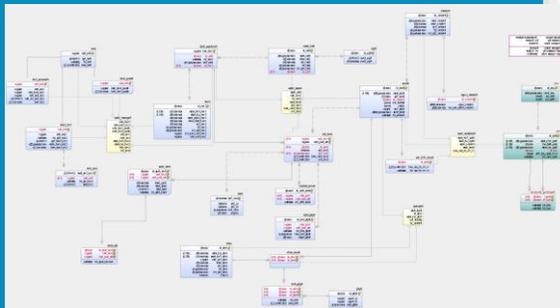
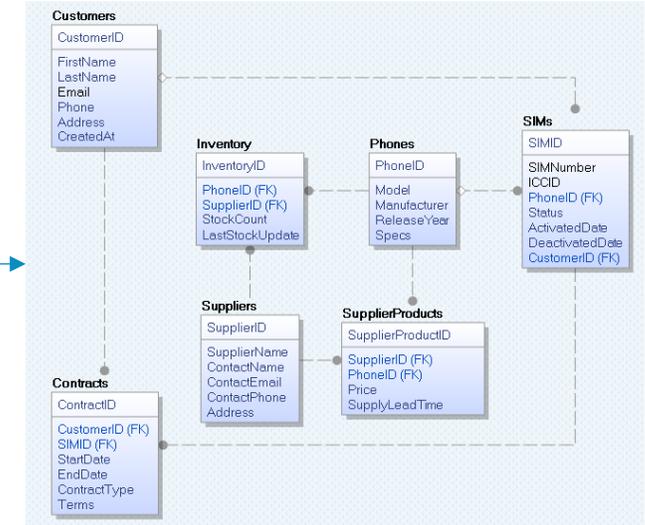
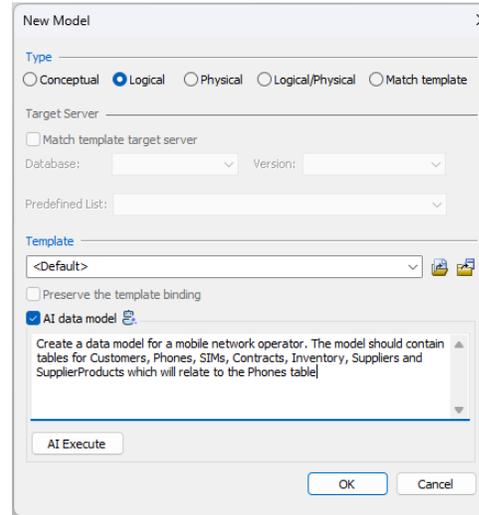




Demo

Data Modeler

- GenAI Logical / Physical models from a prompt
- AI Review physical model schema



```

AI Generated Results. Check for accuracy.
13 ### **Errors**:
```

```

14 ERROR: line 6, column 1, ending line 7, column 26: Found 'CREATE TYPE [ZIP_CODE]
```

```

15 FROM CHAR VARYING(9) NULL': This syntax is only supported as of version 2005
```

```

16
```

```

17 ### **Reasons**:
```

```

18 Valid: 'CHAR VARYING' is not a valid type in SQL Server 2019/2022. Use 'VARCHAR' instead, and omit NULL in CREATE TYPE ... FROM statement.
```

```

19
```

```

20 ### **Corrected Script**:
```

```

21 CREATE TYPE [ZIP_CODE] FROM VARCHAR(9);
```

```

22
```

```

23 ### **Errors**:
```

```

24 ERROR: line 10, column 1, ending line 11, column 22: Found 'CREATE TYPE [STREET_ADDRESS]
```

```

25 FROM VARCHAR(25) NULL': This syntax is only supported as of version 2005
```

```

26
```

```

27 ### **Reasons**:
```

```

28 Valid: Do not specify NULL in CREATE TYPE ... FROM statement for SQL Server. Type syntax is correct when just 'FROM VARCHAR(25)'.
```

```

29
```

```

30 ### **Corrected Script**:
```

```

31 CREATE TYPE [STREET_ADDRESS] FROM VARCHAR(25);
```

```

32
```

```

33 ### **Errors**:
```

```

34 ERROR: line 14, column 1, ending line 15, column 22: Found 'CREATE TYPE [FIRST_NAME]
```

```

35 FROM VARCHAR(15) NULL': This syntax is only supported as of version 2005
```

```

36
```

```

37 ### **Reasons**:
```

```

38 Valid: Do not specify NULL in CREATE TYPE ... FROM statement for SQL Server.
```

```

39
```

```

40 ### **Corrected Script**:
```

```

41 CREATE TYPE [FIRST_NAME] FROM VARCHAR(15);
```

Business Glossary

- Locate business terms requiring curation
- Narrow down the search
- AI generates a description on demand
- Accept / edit / retry / reject
- AI generated descriptions can be searched and reviewed later

erwinAI Assistant

Hello Geoff Hodgkinson! I'm the erwinAI Assistant. How can I assist you today?

Show me business terms missing a description from the Investopedia catalog

There are 621 business glossary terms in the Investopedia catalog missing a description. Please choose how many to display (enter a number between 1 and 20).

#	Name	Catalog	Catalog Path	Description
1	1 Divided 10 Net 30	Investopedia	Banking and Finance/Investopedia	—
2	1040 IRS Form	Investopedia	Banking and Finance/Investopedia	—
3	1040A Form	Investopedia	Banking and Finance/Investopedia	—
4	1040EZ Form	Investopedia	Banking and Finance/Investopedia	—
5	10-K	Investopedia	Banking and Finance/Investopedia	—

Showing: 1-5 of 621

Would you like to generate the **description** for these terms?

yes#

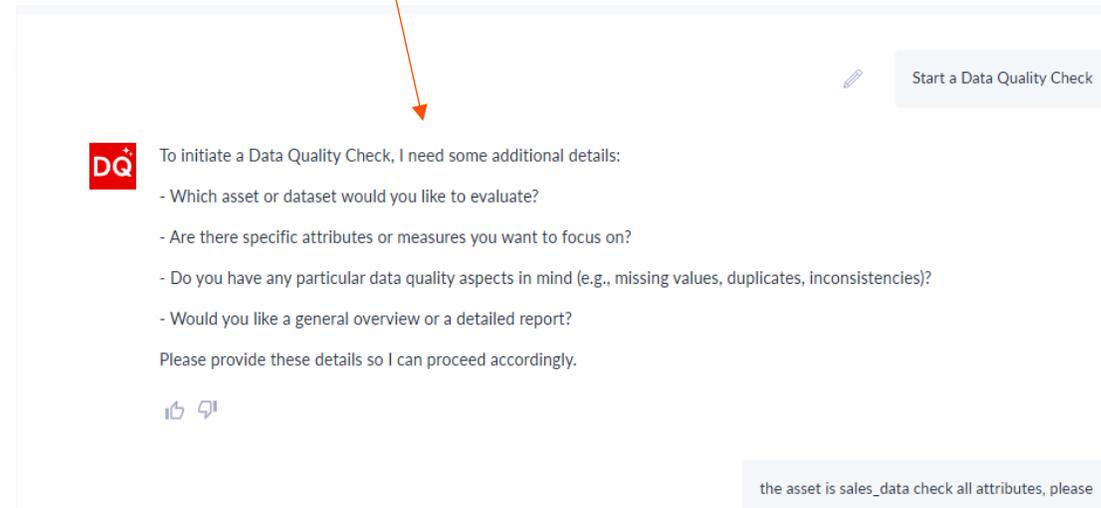
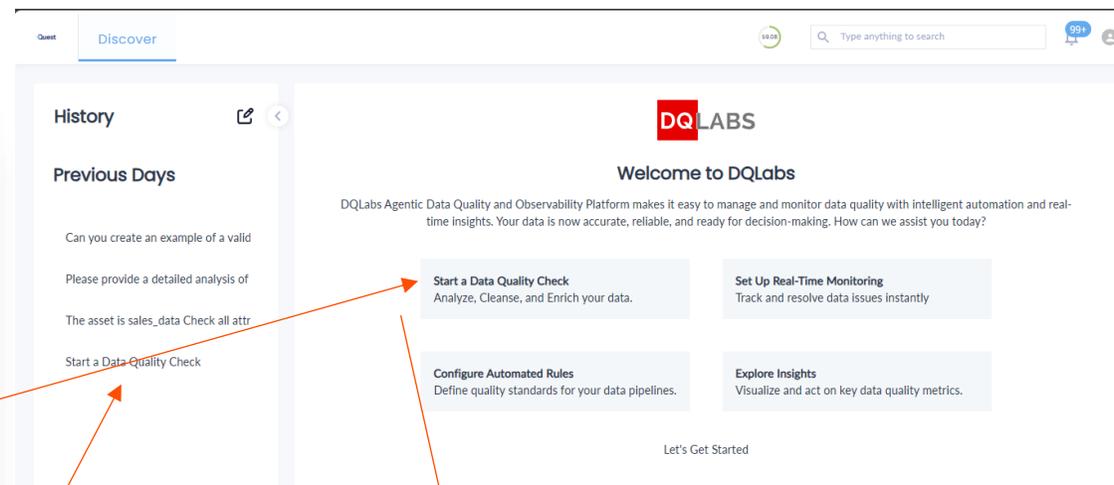
<input type="checkbox"/>	#	Name	Catalog	Catalog Path	Description
<input type="checkbox"/>	1	1 Divided 10 Net 30	Investopedia	Banking and Finance/Investopedia	This term refers to a payment arrangement where a buyer must pay 1% of the invoice amount within 10 days, and the net balance is due within 30 days. It is commonly used in trade credit agreements to incentivize early payment.
<input type="checkbox"/>	2	1040 IRS Form	Investopedia	Banking and Finance/Investopedia	The 1040 IRS Form is the standard federal income tax form used by U.S. taxpayers to file their annual tax returns with the Internal Revenue Service.
<input type="checkbox"/>	3	1040A Form	Investopedia	Banking and Finance/Investopedia	The 1040A Form was a simplified version of the IRS 1040 tax form, used by taxpayers with straightforward tax situations. It is no longer available for use after tax year 2017.
<input type="checkbox"/>	4	1040EZ Form	Investopedia	Banking and Finance/Investopedia	The 1040EZ Form was the simplest IRS tax form for individual taxpayers with basic tax situations. It was discontinued after tax year 2017.
<input type="checkbox"/>	5	10-K	Investopedia	Banking and Finance/Investopedia	The 10-K is an annual report required by the U.S. Securities and Exchange Commission that provides a comprehensive overview of a company's financial performance.

Showing: 1-5 of 621

In the table above, click "Accept" to confirm or "Moderate" to review the entries before finalizing

Data Quality

- Ask AI to find data quality issues
- Ask AI to suggest a measure to track this newly identified issues
- Follow history of previous AI chats



Data Quality (2)

- AI returns some insights
- Further analysis request

the asset is sales_data check all attributes, please



I have retrieved the details for all attributes in the asset "sales_data." Here are some key insights:

- Total attributes analyzed: 20
 - Attributes with issues or alerts: 8
 - Attributes with issues or alerts include:
 - birthdate: 1 alert, 1 issue
 - credit_card_number: 1 alert, 3 issues
 - state: 1 alert, 3 issues
 - id: 1 alert, 3 issues
 - credit_card_type: 1 alert, 2 issues
 - country_code: 2 alerts, 2 issues
 - Attributes with no issues or alerts:
 - last_name, city, first_name, transaction_date, email, zip_code, gender, quantity, price, product
 - Data sizes are consistent across attributes (~393216 units).
 - Data quality scores are generally high, with some attributes like "birthdate," "credit_card_number," "state," "id," "credit_card_type," and "country_code" showing alerts/issues indicating potential data quality concerns.
- Would you like a detailed report on specific attributes or insights into data quality improvement opportunities?



Please provide a detailed analysis of the issues with credit_card_number, including recommendations for measures



Based on the recent alerts and issues identified for the attribute 'credit_card_number', here is a detailed analysis:

Issues Summary:

- Multiple alerts indicate that the credit card numbers are failing constraints related to their expected ranges.
- Specific values such as 1103, 1093, 1083, 1065, 1056, 1047, 1047, 1065, 1056, 1047, 11, and 10 have exceeded the auto constraints or null count limits.
- The constraints for credit card numbers are set within specific ranges, e.g., 1077-1098, 1065-1079, 1048-1069, etc.
- Null counts are also exceeding thresholds, indicating missing or null data entries.

Impact:

- These issues suggest data quality problems such as invalid or inconsistent credit card numbers, which could impact downstream processes like fraud detection, billing, or customer analytics.
- The recurring constraint violations point to potential data entry errors or outdated validation rules.

Recommendations for Measures:

1. Data Validation Measure:

- Create a measure to validate credit card numbers against the expected range and format.
- Example SQL: Check if values fall within the valid ranges and are not null.

2. Null Count Measure:

- Implement a measure to monitor null or missing values in the credit_card_number attribute.

3. Range Constraint Enforcement:

- Establish measures to flag values outside the acceptable ranges for proactive data cleansing.

Data Quality (3)

- Suggested measure
- ... and so on



Can you create an example of a validation rule for credit_card_number which will detect the issues you have identified?



Certainly! Here's an example of a validation rule for the attribute `credit_card_number` that can detect common issues such as invalid formats, incorrect length, and failing the Luhn check:

```
-- Validation Rule for credit_card_number
SELECT *
FROM your_table
WHERE
  -- Check for null or empty values
  credit_card_number IS NULL
  OR credit_card_number = ''

  -- Check for valid credit card format (numeric only, length between 13 and 19)
  OR NOT REGEXP_LIKE(credit_card_number, '^\d{13,19}$')

  -- Check for invalid checksum using Luhn Algorithm
  OR NOT luhn_check(credit_card_number)
```

Note: The `luhn_check` function is a custom or built-in function depending on your database system that validates the credit card number checksum using the Luhn algorithm.

Would you like me to generate the SQL code for the `luhn_check` function as well?



Automated Data Product Factory

- Natural language data product creation
- AI-generated results
- Refine the data product
- AI suggested enhancements
- Data assets matched to cataloged assets = green header
- Data assets generated by AI = grey header

The screenshot displays the Quest Trusted Data Management Platform interface. The top navigation bar includes the Quest logo, the platform name, and system status. The main content area shows a data model for 'Credit Card Fraud Monitoring' with several entities: Person, Person Credit Card, Credit Card, Customer Account, Transactions, and Customer Service Case. Each entity is represented by a table with its attributes and data types. The 'Person' entity is highlighted with a green header, indicating it is a cataloged asset, while others are grey. A sidebar on the left contains navigation options like Home, Modeling, Catalog, Market, Govern, Quality, Product, and Settings. On the right, the 'Automated Data Product Factory' chat interface is open, showing a conversation where a user asks 'I want to track credit card fraud'. The system responds with a summary of the created data product and offers suggestions for further enhancements, such as adding fraud alert and detection event entities. A text input field at the bottom of the chat allows for further refinement of the data product.

Quest Trusted Data Management Platform

All Systems Operational

Geoff.Hodgkinson@quest.com

Data Products > Credit Card Fraud Monitoring

More Actions Publish

Dataset

Person

Person Identifier	IDENTIFIER (PK) <PK>
Person Type Code	CODE
Name Style Indicator	BOOLEAN
Title	TEXT
First Name	TEXT
Middle Name	TEXT
Last Name	TEXT
Name Suffix	TEXT
Email Promotion Preference Code	CODE
Additional Contact Information	TEXT (LONG)
Demographic Details	TEXT (LONG)
Person Global Unique Identifier	TEXT
Last Modified Date	DATETIME

6 Logical Entities PII

Person Credit Card

Credit Card Identifier	IDENTIFIER (PK) <PK>
Person Identifier	IDENTIFIER (FK) <FK>
Relationship Last Modified Date	DATETIME

Credit Card

Card Type	TEXT
Card Number	TEXT
Expiration Month	NUMBER (INTEGER)
Expiration Year	NUMBER (YEAR)
Last Modified Date	DATETIME
Credit Card Identifier_1	IDENTIFIER (FK) <FK>

Customer Account

Account Number	IDENTIFIER (PK) <PK>
Customer Name	TEXT
Customer City	TEXT

Transactions

Transaction Identifier	IDENTIFIER (FK) <FK>
Transaction Date Time	DATETIME
Merchant Name	TEXT
Merchant Category	TEXT
Transaction Amount	NUMBER (DECIMAL)
Cardholder First Name	TEXT
Cardholder Last Name	TEXT

Customer Service Case

Case Identifier	IDENTIFIER (PK) <PK>
Case Number	TEXT
Account Identifier	IDENTIFIER (FK) <FK>
Contact Identifier	IDENTIFIER (FK)
Case Status	CODE

Automated Data Product Factory

Chat Version History

I want to track credit card fraud

Created Credit Card Fraud Monitoring with Transactions, Credit Card, Person Credit Card, Customer Account, and Customer Service Case entities. All relationships established.

Add the Person entity

Person entity was added and related to Person Credit Card via Person Identifier.

Suggestions to further enhance your Credit Card Fraud Monitoring model.

Introduce Fraud Alert and Detection Event entities

Add Merchant and Location reference entities

Model Case Resolution and Dispute Outcome details

Describe the data product you want to create...

0 / 2000

Generated by AI. Verify before using. Governed by <https://quest.com/legal/AI.aspx>

Don't Miss What's Next!

***erwin 15.2: New Features,
AI Enhancements, and
What They Mean for
Modelers***

April 7th @ 11:00a.m. ET



 **Quest**

The Quest logo features a stylized blue icon of a person or figure with arms outstretched, followed by the word "Quest" in a bold, dark blue sans-serif font.