



# Microsoft Implementing Data Engineering Solutions Using Microsoft Fabric DP-700 Prüfungsfragen mit Lösungen (Q53-Q58):

## 53. Frage

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have a Fabric eventstream that loads data into a table named Bike\_Location in a KQL database. The table contains the following columns:

BikepointID  
Street  
Neighbourhood  
No\_Bikes  
No\_Empty\_Docks  
Timestamp

You need to apply transformation and filter logic to prepare the data for consumption. The solution must return data for a neighbourhood named Sands End when No\_Bikes is at least 15. The results must be ordered by No\_Bikes in ascending order.

Solution: You use the following code segment:

```
bike_location
| filter Neighbourhood == "Sands End" and No_Bikes >= 15
| order by No_Bikes
| project BikepointID, Street, Neighbourhood, No_Bikes, No_Empty_Docks, Timestamp
```

Does this meet the goal?

- A. Yes
- B. no

**Antwort: B**

Begründung:

This code does not meet the goal because it uses order by, which is not valid in KQL. The correct term in KQL is sort by.

Correct code should look like:

```
bike_location
| filter Neighbourhood == "Sands End" and No_Bikes >= 15
| sort by No_Bikes asc
| project BikepointID, Street, Neighbourhood, No_Bikes, No_Empty_Docks, Timestamp
```

Topic 1, Litware, IncCase Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Overview

Litware, Inc. is a publishing company that has an online bookstore and several retail bookstores worldwide.

Litware also manages an online advertising business for the authors it represents.

Existing Environment. Fabric Environment

Litware has a Fabric workspace named Workspace1. High concurrency is enabled for Workspace1.

The company has a data engineering team that uses Python for data processing.

Existing Environment. Data Processing

The retail bookstores send sales data at the end of each business day, while the online bookstore constantly provides logs and sales data to a central enterprise resource planning (ERP) system.

Litware implements a medallion architecture by using the following three layers: bronze, silver, and gold. The sales data is ingested from the ERP system as Parquet files that land in the Files folder in a lakehouse.

Notebooks are used to transform the files in a Delta table for the bronze and silver layers. The gold layer is in a warehouse that has V-Order disabled.

Litware has image files of book covers in Azure Blob Storage. The files are loaded into the Files folder.

Existing Environment. Sales Data

Month-end sales data is processed on the first calendar day of each month. Data that is older than one month never changes.

In the source system, the sales data refreshes every six hours starting at midnight each day.

The sales data is captured in a Dataflow Gen1 dataflow. When the dataflow runs, new and historical data is captured. The dataflow captures the following fields of the source:

Sales Date

Author

Price

Units

SKU

A table named AuthorSales stores the sales data that relates to each author. The table contains a column named AuthorEmail.

Authors authenticate to a guest Fabric tenant by using their email address.

Existing Environment. Security Groups

Litware has the following security groups:

Sales

Fabric Admins

Streaming Admins

Existing Environment. Performance Issues

Business users perform ad-hoc queries against the warehouse. The business users indicate that reports against the warehouse sometimes run for two hours and fail to load as expected. Upon further investigation, the data engineering team receives the following error message when the reports fail to load: "The SQL query failed while running." The data engineering team wants to debug the issue and find queries that cause more than one failure.

When the authors have new book releases, there is often an increase in sales activity. This increase slows the data ingestion process.

The company's sales team reports that during the last month, the sales data has NOT been up-to-date when they arrive at work in the morning.

Requirements. Planned Changes

Litware recently signed a contract to receive book reviews. The provider of the reviews exposes the data in Amazon Simple Storage Service (Amazon S3) buckets.

Litware plans to manage Search Engine Optimization (SEO) for the authors. The SEO data will be streamed from a REST API.

Requirements. Version Control

Litware plans to implement a version control solution in Fabric that will use GitHub integration and follow the principle of least privilege.

Requirements. Governance Requirements

To control data platform costs, the data platform must use only Fabric services and items. Additional Azure resources must NOT be provisioned.

Requirements. Data Requirements

Litware identifies the following data requirements:

Process the SEO data in near-real-time (NRT).

Make the book reviews available in the lakehouse without making a copy of the data.

When a new book cover image arrives in the Files folder, process the image as soon as possible.

## 54. Frage

You have a Fabric workspace that contains an eventhouse named Eventhouse1.

In Eventhouse1, you plan to create a table named DeviceStreamData in a KQL database. The table will contain data based on the following sample.

```

Code Segments
:: StreamData:long)
:: .create function EventStreamData (
:: TimeStamp:datetime, DeviceId:string
:: .create table EventStreamData (
:: StreamData:dynamic )

```

Answer Area

Antwort:

Begründung:

Code Segments

```

:: StreamData:long)
:: .create function EventStreamData (
:: TimeStamp:datetime, DeviceId:string
:: .create table EventStreamData (
:: StreamData:dynamic )

```

Answer Area

```

:: .create table EventStreamData (
:: TimeStamp:datetime, DeviceId:string
:: StreamData:long)

```

Explanation:

```

Answer Area
1 :: .create table EventStreamData (
2 :: TimeStamp:datetime, DeviceId:string
3 :: StreamData:long)

```

### 55. Frage

DRAG DROP

You have a Fabric eventhouse that contains a KQL database. The database contains a table named TaxiData.

The following is a sample of the data in TaxiData.

VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	PULocationID	DOLocationID	payment_type	total_amount
2	2022-06-06T11:08:32Z	2022-06-06T11:22:17Z	1	0.17	231	50	2	7.12
2	2022-06-06T11:12:05Z	2022-06-06T11:20:43Z	1	1.02	161	163	1	10.56
1	2022-06-06T11:15:00Z	2022-06-06T11:25:32Z	1	1.07	142	230	2	17.12
2	2022-06-06T11:29:54Z	2022-06-06T11:49:34Z	2	2.07	162	236	2	12.01
1	2022-06-06T11:50:50Z	2022-06-06T12:07:24Z	2	2.65	140	142	1	7.89

You need to build two KQL queries. The solution must meet the following requirements:

One of the queries must partition RunningTotalAmount by VendorID.

The other query must create a column named FirstPickupDateTime that shows the first value of each hour from tpep\_pickup\_datetime partitioned by payment\_type.

How should you complete each query? To answer, drag the appropriate values to the correct targets. Each value may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Values	Answer Area
<input type="checkbox"/> Row_cumsum <input type="checkbox"/> Row_rank_dense <input type="checkbox"/> Row_rank_min <input type="checkbox"/> Row_window_session	<b>Statement1:</b> TaxiData   sort by VendorID asc   extend RunningTotalAmount = 0 (total_amount, VendorID != prev(VendorID))
	<b>Statement2:</b> TaxiData   sort by tpep_pickup_datetime asc, payment_type asc   extend FirstPickupDateTime = 0 (tpep_pickup_datetime, 1h, 0m, payment_type != prev(payment_type))

**Antwort:**

**Begründung:**

Values	Answer Area
<input checked="" type="checkbox"/> Row_cumsum <input type="checkbox"/> Row_rank_dense <input type="checkbox"/> Row_rank_min <input type="checkbox"/> Row_window_session	<b>Statement1:</b> TaxiData   sort by VendorID asc   extend RunningTotalAmount = Row_cumsum (total_amount, VendorID != prev(VendorID))
	<b>Statement2:</b> TaxiData   sort by tpep_pickup_datetime asc, payment_type asc   extend FirstPickupDateTime = Row_window_session (tpep_pickup_datetime, 1h, 0m, payment_type != prev(payment_type))

**Explanation:**

Values	Answer Area
<input checked="" type="checkbox"/> Row_cumsum <input type="checkbox"/> Row_rank_dense <input type="checkbox"/> Row_rank_min <input type="checkbox"/> Row_window_session	<b>Statement1:</b> TaxiData   sort by VendorID asc   extend RunningTotalAmount = Row_cumsum (total_amount, VendorID != prev(VendorID))
	<b>Statement2:</b> TaxiData   sort by tpep_pickup_datetime asc, payment_type asc   extend FirstPickupDateTime = Row_window_session (tpep_pickup_datetime, 1h, 0m, payment_type != prev(payment_type))

Partition the RunningTotalAmount by VendorID. - Row\_cumsum

The Row\_cumsum function computes the cumulative sum of a column while optionally restarting the accumulation based on a condition. In this case, it calculates the cumulative sum of total\_amount for each VendorID, restarting when the VendorID changes (VendorID != prev(VendorID)).

```
TaxiData
| sort by VendorID asc
| extend RunningTotalAmount = Row_cumsum(total_amount, VendorID != prev(VendorID))
```

Create a column FirstPickupDateTime that shows the first value of each hour from tpep\_pickup\_datetime, partitioned by payment\_type - Row\_window\_session

```
TaxiData
| sort by tpep_pickup_datetime asc, payment_type asc
| extend FirstPickupDateTime = Row_window_session(tpep_pickup_datetime, 1h, 0m, payment_type != prev(payment_type))
```

### 56. Frage

You are building a data loading pattern for Fabric notebook workloads. You have the following code segment:

```

def loading_pattern_sample(df_source):
    try:
        deltaTable = DeltaTable.forName(spark, target_table)
    except Exception:
        try:
            df_source.write.format('delta').mode('overwrite').saveAsTable(f"{target_table}")
        except Exception as e:
            print(f'Load for table {target_table} failed with error: {str(e)}')
            raise
        return

    try:
        change_detection_columns = [col for col in df_source.columns if col not in candidate_key]

        match_condition = ' AND '.join([f'target.{col} = source.{col}' for col in candidate_key])
        update_condition = ' OR '.join([f'target.{col} != source.{col}' for col in change_detection_columns])

        update_expr = {col: f'source.{col}' for col in df_source.columns}

        merge_operation = deltaTable.alias('target').merge(
            source=df_source.alias('source'),
            condition=match_condition
        ).whenMatchedUpdate(
            condition=update_condition,
            set=update_expr
        ).whenNotMatchedInsertAll()

        merge_operation.execute()
    except Exception as e:
        print(f'Insert operation for table {target_table} failed with error: {str(e)}')
    return

```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.  
 NOTE: Each correct selection is worth one point.

**ANSWER AREA**

**Statements**

The target table will always be overwritten.

Yes	No
<input type="radio"/>	<input type="radio"/>

The merge operation will always run.

<input type="radio"/>	<input type="radio"/>
-----------------------	-----------------------

The loading pattern supports both full and incremental loading requirements.

<input type="radio"/>	<input type="radio"/>
-----------------------	-----------------------

**Antwort:**

Begründung:

## Answer Area

### Statements

The target table will always be overwritten.

Yes

No

The merge operation will always run.

The loading pattern supports both full and incremental loading requirements.

### 57. Frage

You have a Fabric F32 capacity that contains a workspace. The workspace contains a warehouse named DW1 that is modelled by using MD5 hash surrogate keys.

DW1 contains a single fact table that has grown from 200 million rows to 500 million rows during the past year.

You have Microsoft Power BI reports that are based on Direct Lake. The reports show year-over-year values.

Users report that the performance of some of the reports has degraded over time and some visuals show errors.

You need to resolve the performance issues. The solution must meet the following requirements:

Provide the best query performance.

Minimize operational costs.

Which should you do?

- A. Create views.
- B. Increase the capacity.C Enable V-Order
- C. Modify the surrogate keys to use a different data type.
- D. Change the MD5 hash to SHA256.

**Antwort: A**

Begründung:

In this case, the key issue causing performance degradation likely stems from the use of MD5 hash surrogate keys. MD5 hashes are 128-bit values, which can be inefficient for large datasets like the 500 million rows in your fact table. Using a more efficient data type for surrogate keys (such as integer or bigint) would reduce the storage and processing overhead, leading to better query performance. This approach will improve performance while minimizing operational costs because it reduces the complexity of querying and indexing, as smaller data types are generally faster and more efficient to process.

### 58. Frage

.....

Die Schulungsunterlagen für die Vorbereitung der Microsoft DP-700 Zertifizierungsprüfung beinhalten die Simulationsprüfungen sowie die jetzigen Prüfungsfragen und Antworten zur Microsoft DP-700 Zertifizierungsprüfung. Im Internet haben Sie vielleicht auch einige ähnliche Ausbildungswebsites gesehen. Nach dem Vergleich würden Sie aber finden, dass die Schulungsunterlagen zur Microsoft DP-700 Zertifizierungsprüfung von PrüfungFrage eher zielgerichtet sind. Sie sind nicht nur von guter Qualität, sondern auch die umfassendste.

**DP-700 Pruefungssimulationen:** <https://www.pruefungfrage.de/DP-700-dumps-deutsch.html>

Microsoft DP-700 Kostenlos Downloden Aber sie haben sich nicht am Kurs oder Training im Internet beteiligt, Die von PrüfungFrage DP-700 Pruefungssimulationen bietenden Ressourcen würden Ihre Bedürfnisse sicher abdecken, Neben den hochqualitativen Produkten bieten wir den Käufern den besten zufriedenen Kundenservice an, damit ist unser Microsoft DP-700 Guide in diesem Bereich berühmt, Vielleicht brauchen Sie nur die Zertifizierung der Microsoft DP-700, um Ihren Wunsch des Aufstiegs zu erfüllen.

Eher, als wollten sie einen Kampf verhindern, Erstens erkennen Buchhalter DP-700 an, dass Unternehmen jeder Größe immer globaler werden, Aber sie haben sich nicht am Kurs oder Training im Internet beteiligt.

**Microsoft DP-700: Implementing Data Engineering Solutions Using Microsoft Fabric braindumps PDF & Testking echter Test**

