Optimize a delete operation

Note:

You are not required to complete the processes, tasks, activities, or steps presented in this example. The various samples provided are for illustrative purposes only and it's likely that if you try this out you will encounter issues in your system.

1. Check the number of transaction items for customers with IDs lower than 900000 using the following query:

```
SELECT
    COUNT_BIG(*) as TransactionItemsCount
FROM
    [wwi_perf].[Sale_Hash]
WHERE
    CustomerId < 900000
```

2. DELETE is a fully logged operation. If you need to delete a large amount of data in a table or a partition, it often makes more sense to SELECT the data you wish to keep, which can be run as a minimally logged operation. To select the data, create a new table with CTAS. Once created, use RENAME to swap out your old table with the newly created table. Use the following CTAS query to isolate the transaction items that should be kept:

```
CREATE TABLE [wwi_perf].[Sale_Hash_v2]
WITH
(
    DISTRIBUTION = ROUND_ROBIN,
    HEAP
```

```
)
AS
SELECT
    *
FROM
    [wwi_perf].[Sale_Hash]
WHERE
    CustomerId >= 900000
```

The query should execute within about a minute. All that would remain to complete the process would be to delete the **`Sale_Heap`** table and rename **`Sale_Heap_v2`** to **`Sale_Heap`**.

3. Compare the previous operation with a classical delete:

```
DELETE
    [wwi_perf].[Sale_Hash]
WHERE
    CustomerId < 900000
```

Note The query will run for a potentially long time. Once the time exceeds significantly the time to run the previous CTAS query, you can cancel it (as you can already see the benefit of the CTAS-based approach).