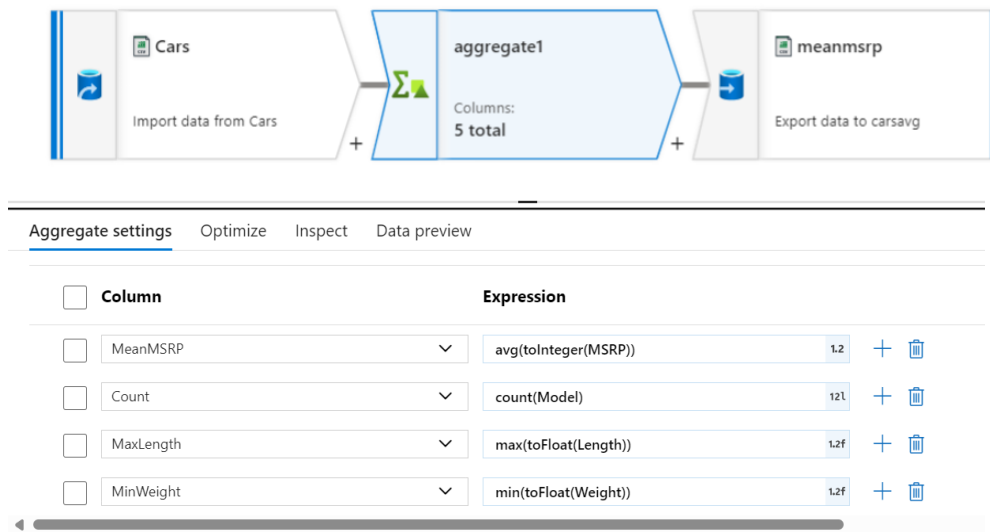# Assignment 10 - Azure Data Factory

1. Design an ADF pipeline to copy data from an on-premise Azure SQL database to Azure Cosmos DB, ensuring data consistency and performance optimization. Pick correct options of partitioning for better performance.
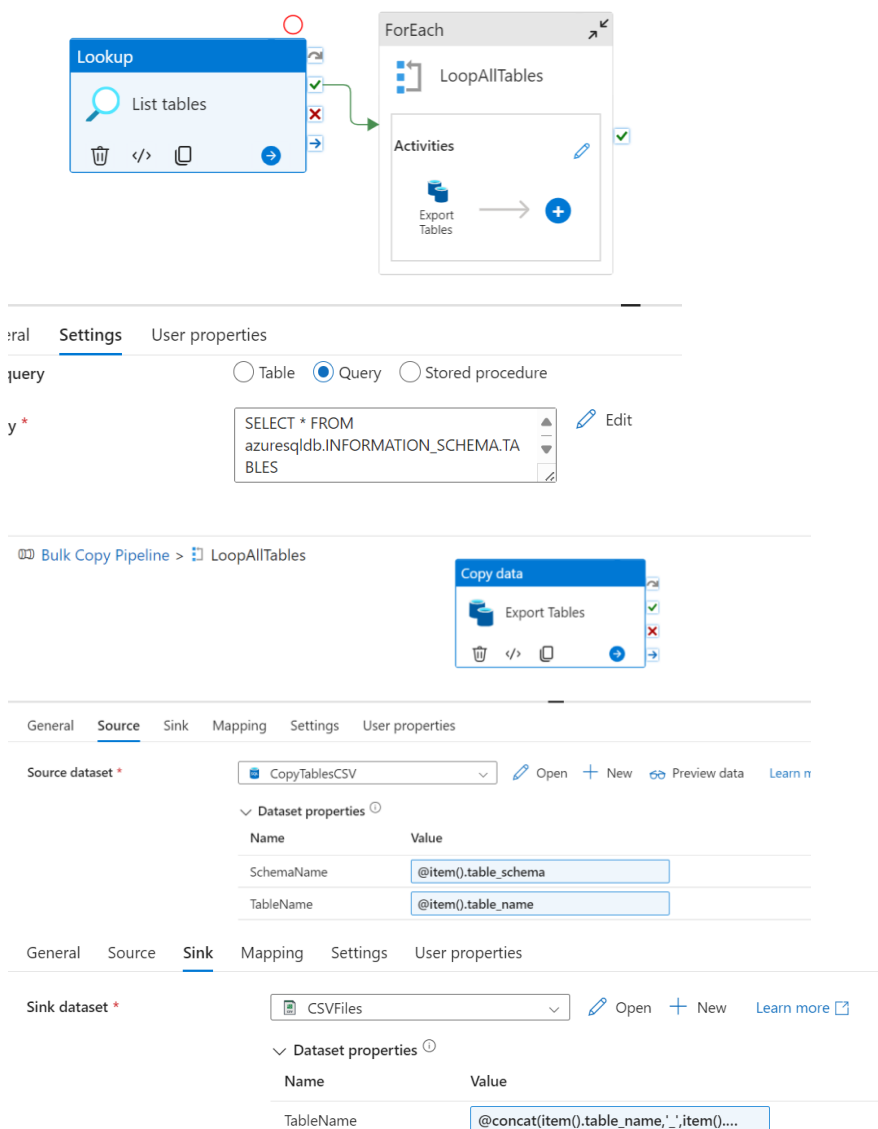

2. Create Pipeline using Azure Data Flow in Azure Data Factory to apply Filter and Sort transformations on datasets.
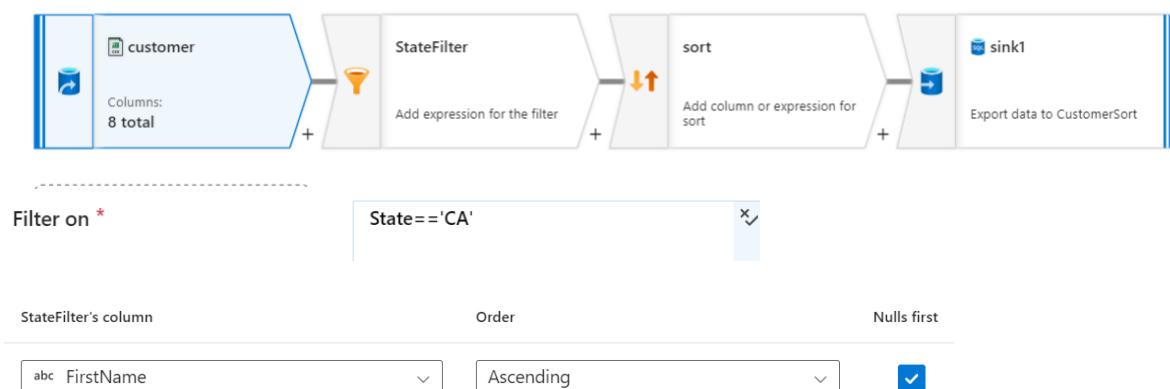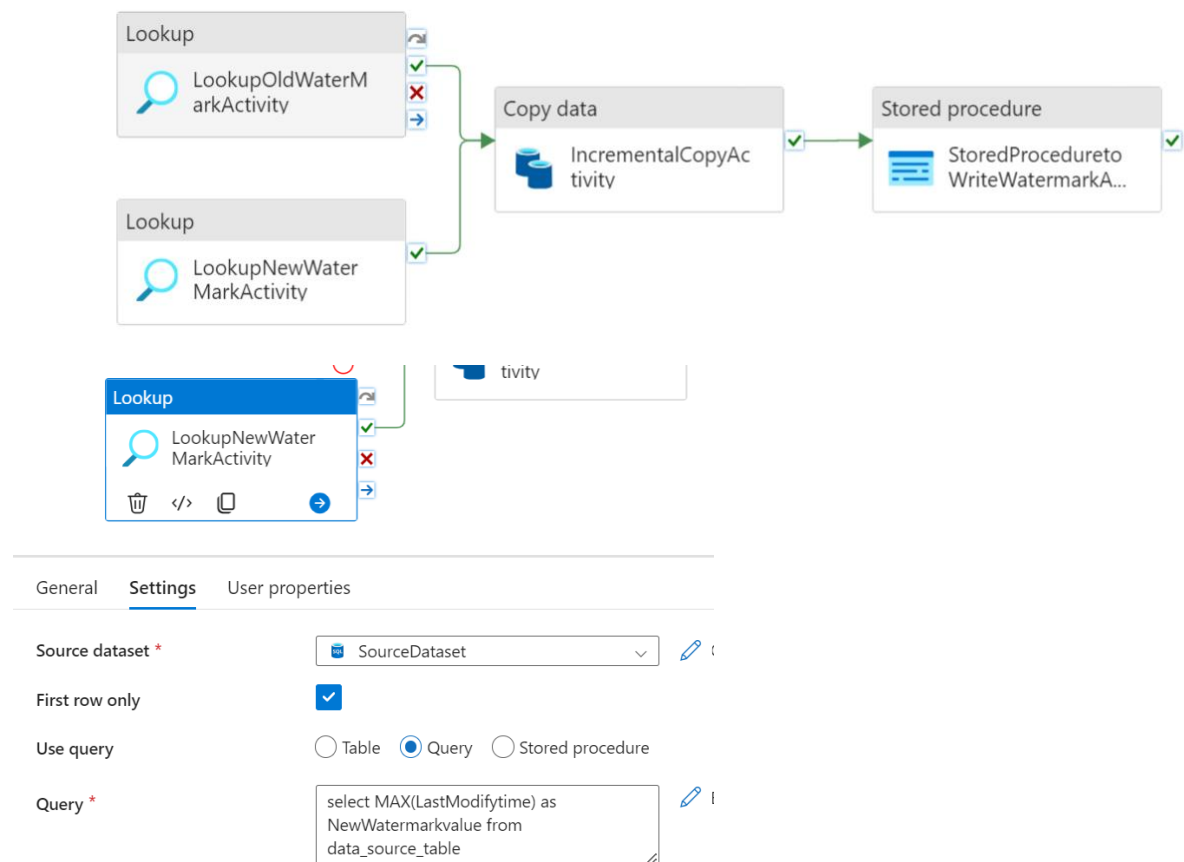

3. Design an ADF pipeline to implement aggregate operations, such as sum, average, max, min and count, within an Azure Data Flow.

Aggregate settings | Optimize | Inspect | Data preview

	Column	Expression	
☐	MeanMSRP ⌄	avg(toInteger(MSRP))	1.2
☐	Count ⌄	count(Model)	12l
☐	MaxLength ⌄	max(toFloat(Length))	1.2f
☐	MinWeight ⌄	min(toFloat(Weight))	1.2f

4. Create best approach to bulk copy data from multiple homogenous sources into Azure SQL Database using ADF pipelines. Show usage of Lookup, For Each Loop and Expressions in Azure Data Factory.

eral | Settings | User properties

query   ◯ Table ⦿ Query ◯ Stored procedure

y *
SELECT * FROM
azuresqldb.INFORMATION_SCHEMA.TA
BLES

✎ Edit

🔗 Bulk Copy Pipeline > ⊟ LoopAllTables

General | Source | Sink | Mapping | Settings | User properties

Source dataset *   CopyTablesCSV ⌄   ✎ Open + New 👓 Preview data Learn m

⌄ Dataset properties ⓘ

Name	Value
SchemaName	@item().table_schema
TableName	@item().table_name

General | Source | Sink | Mapping | Settings | User properties

Sink dataset *   CSVFiles ⌄   ✎ Open + New Learn more ⬚

⌄ Dataset properties ⓘ

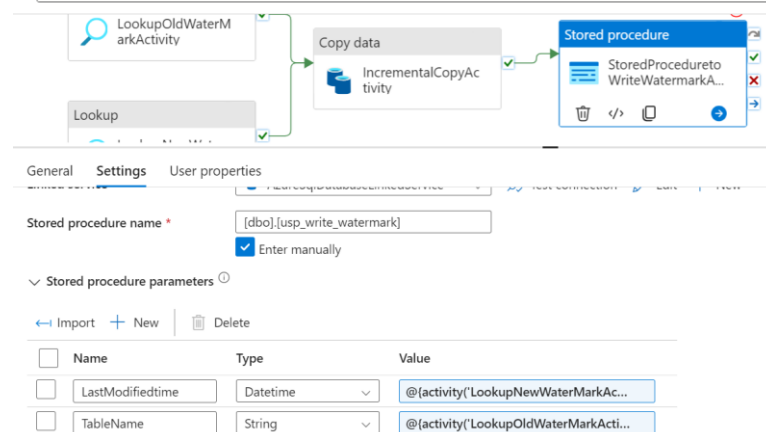| Name	Value
TableName	@concat(item().table_name,'_',item()....

5. Implement incremental load Pipeline in Azure Data Factory for handling datasets, ensuring efficient insert/upsert/updates to the target storage without re-inserting the entire dataset?

Lookup — LookupOldWaterMarkActivity
Lookup — LookupNewWaterMarkActivity
Copy data — IncrementalCopyActivity
Stored procedure — StoredProcedureto WriteWatermarkA...

Lookup — LookupNewWaterMarkActivity

General   **Settings**   User properties

Source dataset *	SourceDataset
First row only	☑
Use query	◯ Table ⦿ Query ◯ Stored procedure
Query *	select MAX(LastModifytime) as NewWatermarkvalue from data_source_table

Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system variabl

```
select * from data_source_table where LastModifytime > '@
    {activity('LookupOldWaterMarkActivity').output.firstRow.
    WatermarkValue}' and LastModifytime <= '@{activity
    ('LookupNewWaterMarkActivity').output.firstRow.
    NewWatermarkvalue}'
```


LookupOldWaterMarkActivity
Copy data — IncrementalCopyActivity
Stored procedure — StoredProcedureto WriteWatermarkA...
Lookup

General   **Settings**   User properties

Stored procedure name *	[dbo].[usp_write_watermark]
	☑ Enter manually

∨ Stored procedure parameters ⓘ

⤙ Import   + New   🗑 Delete

	Name	Type	Value
☐	LastModifiedtime	Datetime	@{activity('LookupNewWaterMarkAc...
☐	TableName	String	@{activity('LookupOldWaterMarkActi...

6. What are the key steps to connect Azure Databricks to Cosmos DB for real-time analytics and data transformation using spark and Databricks.

- Create a Databricks Workspace in Azure.
- Set up a Cluster in the Databricks workspace.
- Install the Cosmos DB Spark Connector on the cluster:
  For Spark 3.x: Use azure-cosmos-spark_3-1_2-12 (from Maven).
- Create a Cosmos DB Account in Azure.
- Generate Cosmos DB Connection Details:
  Get the URI, Primary Key, and Database Name from your Cosmos DB account.
- Write Connection Code in Databricks:
  Use Spark with the Cosmos DB connector to establish the connection:

```
cosmosConfig = {
"Endpoint": "your-cosmosdb-uri",
"Masterkey": "your-cosmosdb-primary-key",
"Database": "your-cosmosdb-database-name",
"Collection": "your-collection-name"
}

df = spark.read.format("cosmos.oltp").options(**cosmosConfig).load()
```

- Perform Data Transformations in Databricks using Spark on the Cosmos DB data.
- Write Data Back to Cosmos DB (optional):
  Use .write() method to send processed data back to Cosmos DB.