**PROBLEM:**

* Store and analyse data from devices and sensors in HVAC and lighting systems
* Differentiate Database and Data Warehouse Design
* Ensure scalability and efficiency

**SOLUTION APPROACH:**

* **Database:** Creating tables with relationships and using Azure SQL Database for scalability and managed services
* **Data Warehousing:** Creating tables with no relationships, utilize Azure Synapse Analytics for serverless capabilities and seamless integration with Azure SQL.
* **Data Model:** Normalized schema (Star) to optimize storage and querying
* **Migrations:** Flyway for version control and automated deployment

**SCALABILITY:**

* Sharding & partitioning
* Archiving

**DATABASE SCHEMA:**

Device Table:

CREATE TABLE Device

(

DId INT IDENTITY(1,1),

DeviceId UNIQUEIDENTIFIER PRIMARY KEY,

DeviceName NVARCHAR(255) NOT NULL,

CreatedAt DATETIME NOT NULL

);

DeviceReading Table:

CREATE TABLE DeviceReading

(

[DId] INT IDENTITY(1,1),

[DeviceId] UNIQUEIDENTIFIER NOT NULL,

[CurrentValue] FLOAT NOT NULL,

[Unit] VARCHAR(50) NOT NULL,

[ReadingTimestamp] DATETIMEOFFSET NOT NULL,

[Version] FLOAT NOT NULL,

FOREIGN KEY (DeviceId) REFERENCES Device(DeviceId)

);

**QUERY PLAN:**

**Index for faster efficiency and query performance:**

CREATE INDEX idx\_deviceId\_timestamp ON DeviceReading (DeviceId, ReadingTimestamp);

**Data Retrieval Query:**

WITH LatestReadings AS (

SELECT DeviceId, MAX(ReadingTimestamp) AS ReadingTimestamp

FROM DeviceReading

GROUP BY DeviceId

)

SELECT

dr.DeviceId,

d.DeviceName,

dr.CurrentValue,

dr.Unit,

dr.ReadingTimestamp

FROM DeviceReading dr

JOIN Device d ON dr.DeviceId = d.DeviceId

JOIN LatestReadings lr ON dr.DeviceId = lr.DeviceId AND dr.ReadingTimestamp = lr.ReadingTimestamp;

**DATA WAREHOUSING DESIGN:**

CREATE SCHEMA DW;

**Schema for Data Warehousing (Star):**

**Device Dimension Table**

CREATE TABLE DW.d\_Device

(

[SK] INT IDENTITY (1, 1),

[DeviceId] UNIQUEIDENTIFIER NOT NULL UNIQUE,

[DeviceName] NVARCHAR(255) NOT NULL,

[CreatedAt] DATETIMEOFFSET NOT NULL,

[ETLLoadDate] DATETIME NOT NULL,

[UpdatedAt] DATETIME NULL

);

**DeviceReading Fact Table**

CREATE TABLE DW.f\_DeviceReading

(

[SK] INT IDENTITY (1, 1),

[DeviceSK] INT NOT NULL,

[CurrentValue] FLOAT NOT NULL,

[Unit] VARCHAR(50) NOT NULL,

[ReadingTimestamp] DATETIMEOFFSET NOT NULL,

[Version] FLOAT NOT NULL,

[ETLLoadDate] DATETIME NOT NULL,

[UpdatedAt] DATETIME NULL

);

**QUERY PLAN (DW):**

**Index for faster efficiency and query performance:**

CREATE CLUSTERED INDEX IDX\_DeviceSK ON DW.f\_DeviceReading (DeviceSK);

CREATE CLUSTERED INDEX IX\_d\_Device\_SK ON DW.d\_Device(SK);

**Data Retrieval Query:**

SELECT

d.DeviceName,

dr.CurrentValue,

dr.ReadingTimestamp,

dr.Unit

FROM

DW.f\_DeviceReading dr

JOIN

DW.d\_Device d ON d.SK = dr.DeviceSK ;

**BIG DATA INTEGRATION:**

1. Storing raw and processed data in Azure Data Lake Storage (ADLS) for further processing
2. Realtime Analytics using stream analytics
3. Use Databricks for advanced analytics and processing large volume of data

**PIPELINE DESIGN:**

1. **Source to ADLS**

* Store data in original format in ADLS for further processing

1. **ADLS to Azure Synapse (Dimension and Fact Tables)**

* Load and transform data from ADLS into dimension and fact tables in Azure Synapse Analytics directly using Dataflow.

1. **Optional Middle Staging**

* Use intermediate staging tables in Azure Synapse for processing data before loading into the data warehouse (DW)

1. **Audit Logging**
   * To log or track errors and status of each pipeline