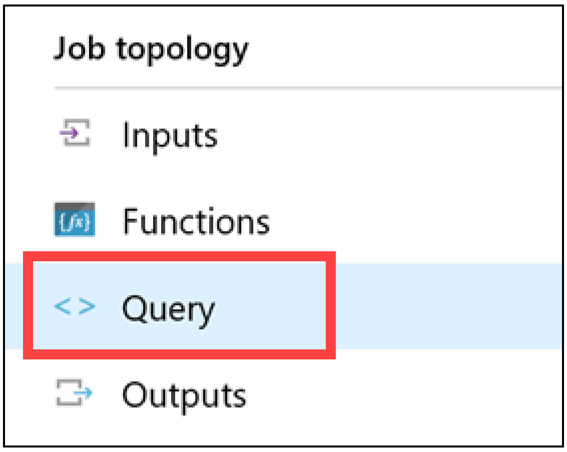
Exercise – Create a query using tumbling windows

8 minutes

In this exercise, you create a Synapse Analytics query using a Tumbling Window. The query will aggregate streaming data received from the Event Hub input and send it to Power BI and Azure Synapse Analytics for visualization and analysis.

1. From your Stream Analytics job's blade in the Azure portal, select **Query** in the left-hand navigation menu.

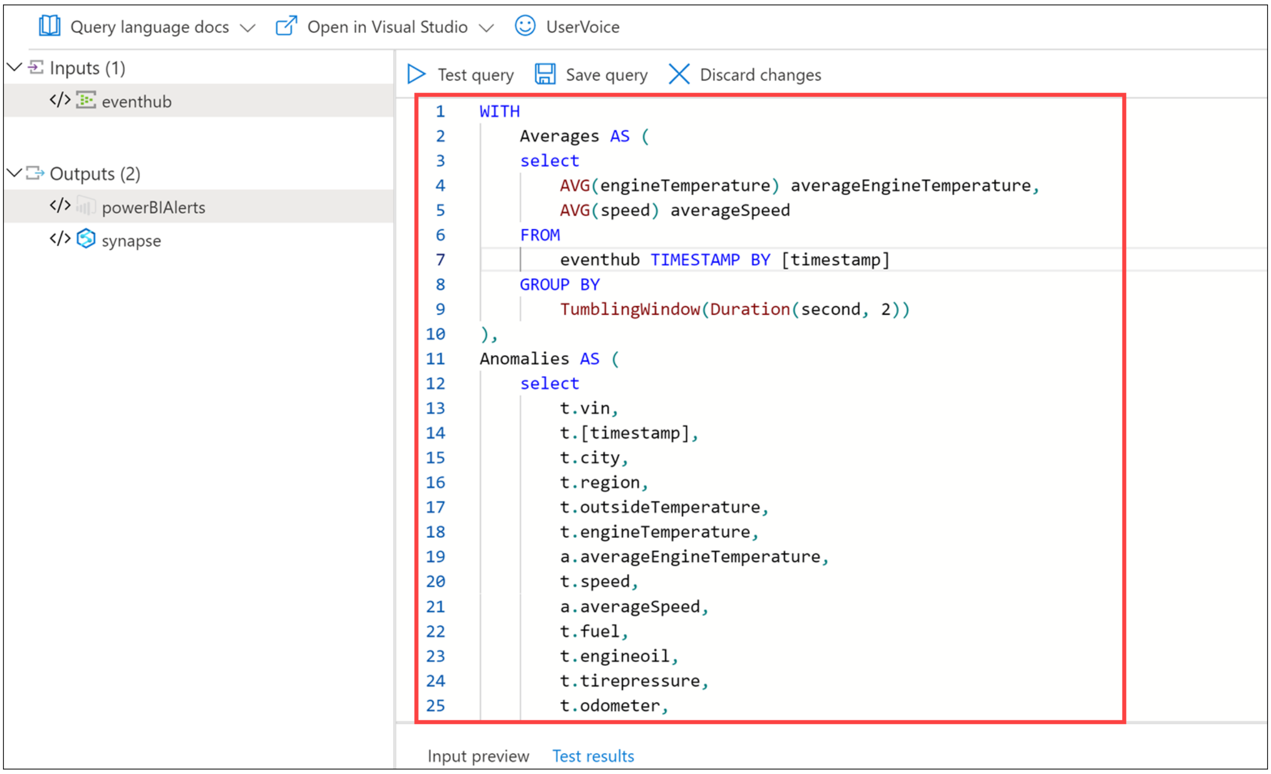
2. Clear the edit **Query** window and paste the following in its place:

SQL

```sql
WITH Averages AS (
    SELECT
        AVG(engineTemperature) averageEngineTemperature,
        AVG(speed) averageSpeed
    FROM
        eventhub TIMESTAMP BY [timestamp]
    GROUP BY
        TumblingWindow(Duration(second, 2))
),
Anomalies AS (
    select
        t.vin,
        t.[timestamp],
        t.city,
        t.region,
        t.outsideTemperature,
        t.engineTemperature,
        a.averageEngineTemperature,
        t.speed,
        a.averageSpeed,
        t.fuel,
        t.engineoil,
        t.tirepressure,
        t.odometer,
        t.accelerator_pedal_position,
        t.parking_brake_status,
        t.headlamp_status,
        t.brake_pedal_status,
        t.transmission_gear_position,
        t.ignition_status,
        t.windshield_wiper_status,
        t.abs,
        (CASE WHEN a.averageEngineTemperature >= 405 OR a.averageEngineTemperature <= 15 THEN 1 ELSE 0 END) AS enginetempanomaly,
        (CASE WHEN t.engineoil <= 1 THEN 1 ELSE 0 END) AS oilanomaly,
        (CASE WHEN (t.transmission_gear_position = 'first' OR
            t.transmission_gear_position = 'second' OR
            t.transmission_gear_position = 'third') AND
            t.brake_pedal_status = 1 AND
            t.accelerator_pedal_position >= 90 AND
            a.averageSpeed >= 55 THEN 1 ELSE 0 END) AS aggressivedriving
    FROM eventhub t TIMESTAMP BY [timestamp]
    INNER JOIN Averages a ON DATEDIFF(second, t, a) BETWEEN 0 And 2
),
VehicleAverages AS (
    SELECT
        AVG(engineTemperature) averageEngineTemperature,
        AVG(speed) averageSpeed,
        System.TimeStamp() AS snapshot
    FROM
        eventhub TIMESTAMP BY [timestamp]
    GROUP BY
        TumblingWindow(Duration(minute, 2))
)

-- INSERT INTO POWER BI
SELECT
    *
INTO
    powerBIAlerts
FROM
    Anomalies
WHERE aggressivedriving = 1 OR enginetempanomaly = 1 OR oilanomaly = 1

-- INSERT INTO SYNAPSE ANALYTICS
SELECT
    *
INTO
    synapse
FROM
    VehicleAverages
```



```
WITH
    Averages AS (
    select
        AVG(engineTemperature) averageEngineTemperature,
        AVG(speed) averageSpeed
    FROM
        eventhub TIMESTAMP BY [timestamp]
    GROUP BY
        TumblingWindow(Duration(second, 2))
),
Anomalies AS (
    select
        t.vin,
        t.[timestamp],
        t.city,
        t.region,
        t.outsideTemperature,
        t.engineTemperature,
        a.averageEngineTemperature,
        t.speed,
        a.averageSpeed,
        t.fuel,
        t.engineoil,
        t.tirepressure,
        t.odometer,
```

The query averages the engine temperature and speed over a two-second duration by adding `TumblingWindow(Duration(second, 2))` to the query's `GROUP BY` clause. Then it selects all telemetry data, including the average values from the previous step, and specifies the following anomalies as new fields:

a. **enginetempanomaly**: When the average engine temperature is >= 405 or <= 15.

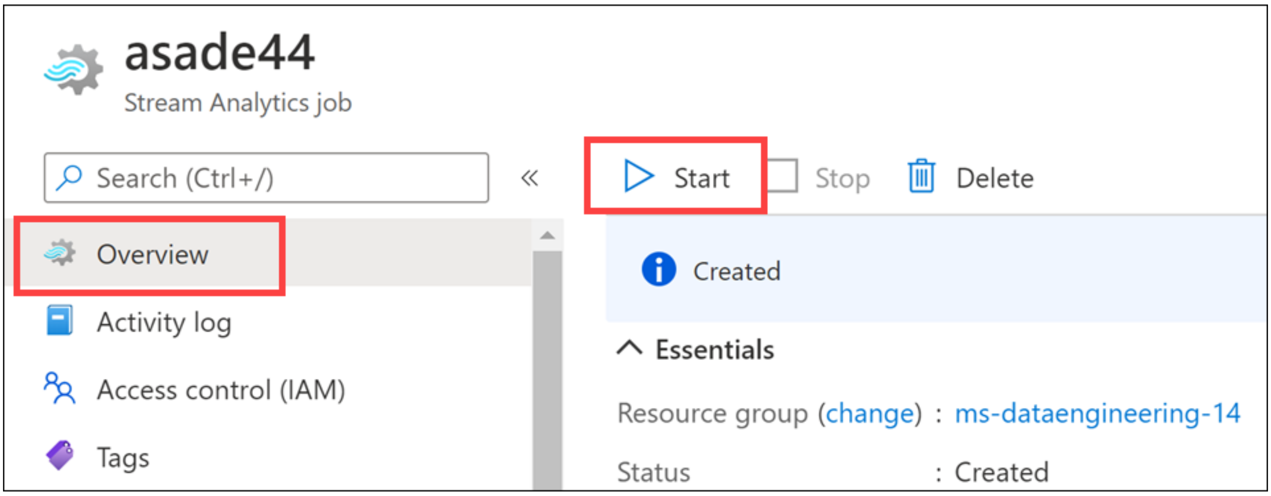
b. **oilanomaly**: When the engine oil <= 1.

c. **aggressivedriving**: When the transmission gear position is in first, second, or third, and the brake pedal status is 1, the accelerator pedal position >= 90, and the average speed is >= 55.

The query outputs all fields from the anomalies step into the `powerBIAlerts` output where `aggressivedriving = 1` or `enginetempanomaly = 1` or `oilanomaly = 1` for reporting. The query also aggregates the average engine temperature and speed of all vehicles over the past two minutes, using `TumblingWindow(Duration(minute, 2))`, and outputs these fields to the `synapse` output.

3. Select **Save query** in the top toolbar when you have finished updating the query.

4. To start the query, select **Overview** within the Stream Analytics job blade's left-hand navigation menu. On top of the Overview blade, select **Start**.



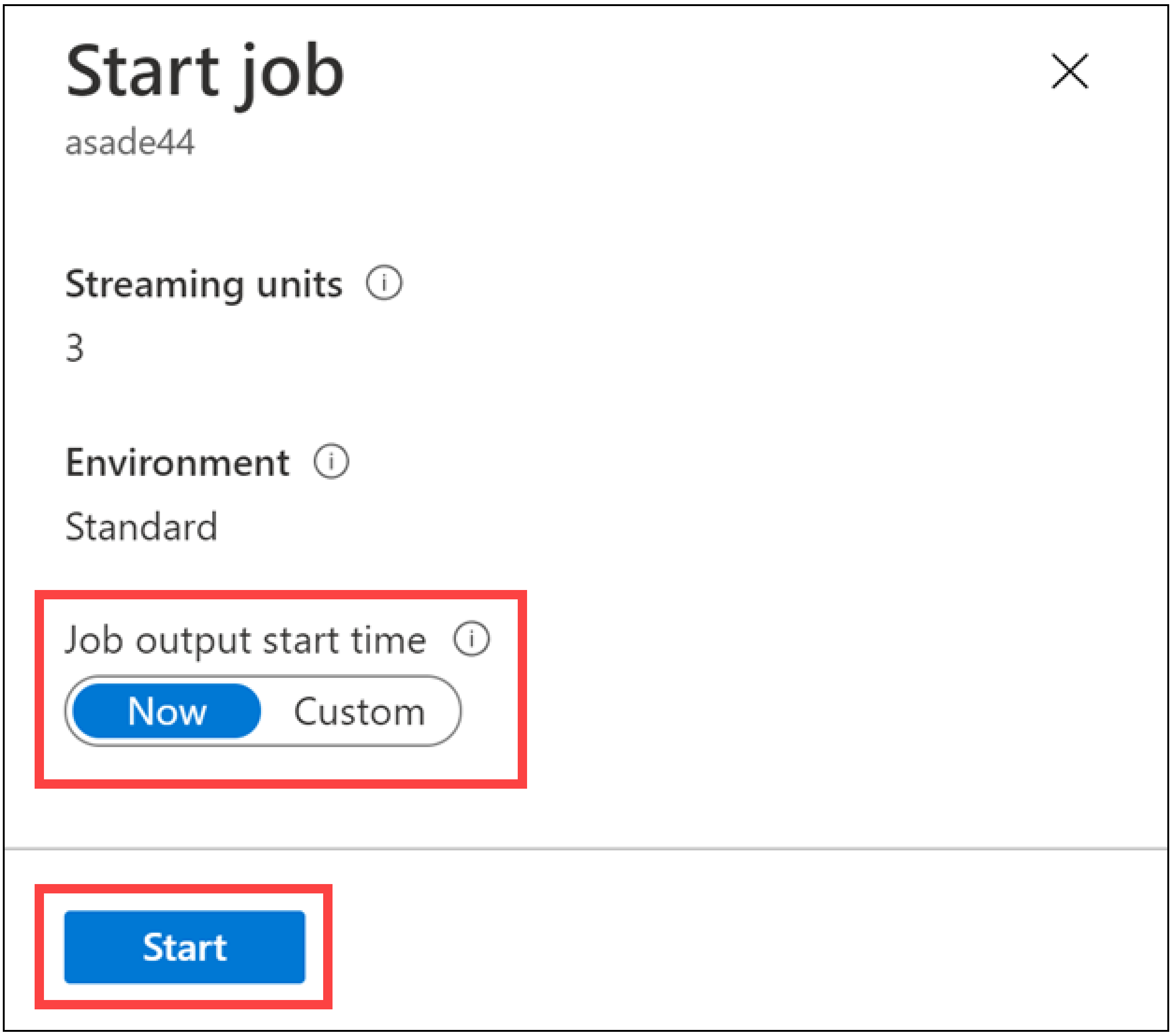
asade44

Stream Analytics job

Resource group (change) : ms-dataengineering-14

Status : Created

5. In the Start job blade that appears, select **Now** for the job output start time, then select **Start**. This will start the Stream Analytics job, so it will be ready to start processing and sending your events to Power BI later on.



Start job

asade44

Streaming units: 3

Environment: Standard

Job output start time: [✓ Now] Custom