```
    -- Create a new database
CREATE DATABASE CUSTOMER_INSIGHTS_DATA;

-- Switch to the new database
USE CUSTOMER_INSIGHTS_DATA;

-- Create the Customer_Personal_Data table
CREATE TABLE Customer_Personal_Data (
    Customer_ID NVARCHAR(10) PRIMARY KEY,
    Customer_Name NVARCHAR(100),
    Customer_Email NVARCHAR(255),
    Age INT,
    Gender NVARCHAR(10)
);

-- Bulk insert data into the table from a CSV file
BULK INSERT Customer_Personal_Data
FROM 'D:\Vertocity\END CAPSTONES\END CAPSTONE 1\CUSTOMER INSIGHTS DATASET\Customer Personal Data.csv'
WITH
(
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2
);

-- Verify the inserted data
SELECT *
FROM Customer_Personal_Data;


-- Create the Customer_Sales_Data table with a foreign key reference
CREATE TABLE Customer_Sales_Data (
    Customer_ID NVARCHAR(10),
    Customer_Name NVARCHAR(100),
    Customer_Email NVARCHAR(255),
    Age_Group NVARCHAR(50),
    Total_Purchases INT,
    Amount_Spent DECIMAL(18, 2),
    Customer_Since INT,
    FOREIGN KEY (Customer_ID) REFERENCES Customer_Personal_Data(Customer_ID)
);

-- Bulk insert data into the table from a CSV file
BULK INSERT Customer_Sales_Data
FROM 'D:\Vertocity\END CAPSTONES\END CAPSTONE 1\CUSTOMER INSIGHTS DATASET\Customer Sales Data.csv'
WITH
(
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2
);
```

```sql
-- Verify the inserted data
SELECT *
FROM Customer_Sales_Data;


-- Let's proceed with the deliverables

/* 1. Execute an INNER JOIN between the two tables based on Customer_ID. */

SELECT *
FROM
    Customer_Personal_Data AS CPD
INNER JOIN
    Customer_Sales_Data AS CSD
ON
    CPD.Customer_ID = CSD.Customer_ID;

/* 2. Find customers who became members in 2020 or later */

SELECT *
FROM
    Customer_Personal_Data AS CPD
INNER JOIN
    Customer_Sales_Data AS CSD
ON
    CPD.Customer_ID = CSD.Customer_ID
WHERE
    CSD.Customer_Since >= 2020
ORDER BY
    CSD.Customer_Since ASC;

/* 3. Create a function to calculate the average purchase value for a given customer ID.
      The function should return NULL or a default value if there are no purchases. */

CREATE FUNCTION calculate_avg_purchase_value (@CustomerID NVARCHAR(10))
RETURNS FLOAT
AS
BEGIN
    DECLARE @AvgPurchaseValue FLOAT;

    SELECT @AvgPurchaseValue = AVG(CAST(Amount_Spent AS FLOAT))
    FROM Customer_Sales_Data
    WHERE Customer_ID = @CustomerID;

    RETURN ISNULL(@AvgPurchaseValue, 0);
END;

/* Positive Test Cases */

SELECT dbo.calculate_avg_purchase_value('CID0025') AS 'Average Purchase Amount';
SELECT dbo.calculate_avg_purchase_value('CID0195') AS 'Average Purchase Amount';

/* Negative Test Cases */

SELECT dbo.calculate_avg_purchase_value('CID0297') AS 'Average Purchase Amount';
SELECT dbo.calculate_avg_purchase_value('CID0360') AS 'Average Purchase Amount';

/* 4. Create a view to categorize customers as 'High Spenders', 'Moderate Spenders',
      or 'Low Spenders' based on their total amount spent. */

CREATE VIEW segmented_customers AS
SELECT
    CPD.Customer_ID,
    CPD.Customer_Name,
    CPD.Customer_Email,
    SUM(CSD.Amount_Spent) AS Total_Spent,
    CASE
        WHEN SUM(CSD.Amount_Spent) <= 2000 THEN 'Low Spenders'
        WHEN SUM(CSD.Amount_Spent) > 2000 AND SUM(CSD.Amount_Spent) <= 5000 THEN 'Moderate Spenders'
        ELSE 'High Spenders'
    END AS Spending_Category
FROM
    Customer_Personal_Data CPD
INNER JOIN
    Customer_Sales_Data CSD
ON
    CPD.Customer_ID = CSD.Customer_ID
GROUP BY
    CPD.Customer_ID, CPD.Customer_Name, CPD.Customer_Email;

/* Verify the newly created view */

SELECT *
FROM segmented_customers
ORDER BY
    CASE
        WHEN Spending_Category = 'Low Spenders' THEN 1
        WHEN Spending_Category = 'Moderate Spenders' THEN 2
        WHEN Spending_Category = 'High Spenders' THEN 3
    END DESC;


-- THE END --
```