```sql
SELECT * FROM [HR-Employee-Attrition];


-- OVERALL ATTRITION RATE

SELECT COUNT(*) AS total_employees,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) AS attrition_count,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) * 100.0 / COUNT(*) AS
         attrition_rate
FROM [HR-Employee-Attrition];


-- ATTRITION RATES BY AGE GROUP

SELECT Age,
       COUNT(*) AS total_employees,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) AS attrition_count,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) * 100.0 / COUNT(*) AS
         attrition_rate
FROM [HR-Employee-Attrition]
GROUP BY Age
ORDER BY Age;


-- ATTRITION RATES BY GENDER

SELECT Gender,
       COUNT(*) AS total_employees,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) AS attrition_count,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) * 100.0 / COUNT(*) AS
         attrition_rate
FROM [HR-Employee-Attrition]
GROUP BY Gender
ORDER BY Gender;


-- ATTRITION RATES BY MARITAL STATUS

SELECT MaritalStatus,
       COUNT(*) AS total_employees,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) AS attrition_count,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) * 100.0 / COUNT(*) AS
         attrition_rate
FROM [HR-Employee-Attrition]
GROUP BY MaritalStatus
ORDER BY MaritalStatus;


-- ATTRITION RATES BY JOB LEVEL

SELECT JobLevel,
```

```sql
       COUNT(*) AS total_employees,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) AS attrition_count,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) * 100.0 / COUNT(*) AS
         attrition_rate
FROM [HR-Employee-Attrition]
GROUP BY JobLevel
ORDER BY JobLevel;


-- ATTRITION RATES BY DEPARTMENT

SELECT Department,
       COUNT(*) AS total_employees,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) AS attrition_count,
       SUM(CASE WHEN Attrition = '1' THEN 1 ELSE 0 END) * 100.0 / COUNT(*) AS
         attrition_rate
FROM [HR-Employee-Attrition]
GROUP BY Department
ORDER BY Department;


-- PERFORMANCE RATING DISTRIBUTION FOR EMPLOYEES WHO LEFT

SELECT PerformanceRating,
       COUNT(*) AS attrition_count
FROM [HR-Employee-Attrition]
WHERE Attrition = '1'
GROUP BY PerformanceRating
ORDER BY PerformanceRating;


-- PERFORMANCE RATING DISTRIBUTION FOR EMPLOYEES WHO STAYED

SELECT PerformanceRating,
       COUNT(*) AS no_attrition_count
FROM [HR-Employee-Attrition]
WHERE Attrition = '0'
GROUP BY PerformanceRating
ORDER BY PerformanceRating;


-- FACTORS INFLUENCING ATTRITION

SELECT
    Age,
    Gender,
    PerformanceRating,
    COUNT(*) AS attrition_count
FROM [HR-Employee-Attrition]
WHERE Attrition = '1'
GROUP BY Age, Gender, PerformanceRating
ORDER BY attrition_count DESC;
```

```sql
-- SPLIT THE DATASET INTO TRAINING AND TESTING (70:30 SPLIT)

SELECT *
INTO train_data
FROM [HR-Employee-Attrition]
TABLESAMPLE SYSTEM(70);

SELECT *
INTO test_data
FROM [HR-Employee-Attrition]
WHERE Employee_ID NOT IN (SELECT Employee_ID FROM train_data);

SELECT
    Employee_ID,
    Age,
    Gender,
    JobRole,
    PerformanceRating,
    CASE WHEN attrition = '1' THEN 1 ELSE 0 END AS attrition_label,
    ROW_NUMBER() OVER (PARTITION BY attrition ORDER BY NEWID()) AS row_num
INTO employee_attrition_model
FROM [HR-Employee-Attrition];


-- CREATE A LOGISTIC REGRESSION MODEL

SELECT
    Employee_ID,
    Age,
    Gender,
    JobRole,
    PerformanceRating,
    attrition_label,
    ROW_NUMBER() OVER (PARTITION BY attrition_label ORDER BY NEWID()) AS row_num
INTO employee_attrition_model_trained
FROM employee_attrition_model
WHERE row_num <= 0.7 * (SELECT COUNT(*) FROM employee_attrition_model);
```