```
CREATE TABLE Employee_perf (
    employee_id INT NOT NULL,
    Attrition TEXT,
    HourlyRate INT,
    JobLevel INT,
    JobSatisfaction   INT,
    MonthlyIncome INT,
    MonthlyRate      INT,
    PerformanceRating INT,
 PRIMARY KEY (employee_id)
 );


CREATE TABLE Employee_test (
    employee_id  INT NOT NULL,
    department TEXT,
    region TEXT,
    education TEXT,
    gender    TEXT,
    recruitment_channel       TEXT,
    no_of_trainings INT,
    age         INT,
    previous_year_rating INT,
    length_of_service INT,
    awards_won INT,
    avg_training_score INT,
PRIMARY KEY (employee_id)
);


-- 1. How many employees do we have in the organization and what is the maximum length of
service?

SELECT COUNT(employee_id) AS TotalEmployees, MAX(length_of_service) AS
MaxServiceLength
FROM employee_test;

-- 2. How many employees are there in each department?
SELECT department, COUNT(employee_id) AS EmployeeCount
FROM employee_test
GROUP BY department;

-- 3. What is the proportion of male to female employees?
SELECT gender, COUNT(employee_id) AS EmployeeCount
FROM employee_test
GROUP BY gender;

SELECT
```

```
    gender,
    COUNT(*) AS count,
    ROUND((COUNT(*) * 100.0 / (SELECT COUNT(*) FROM employee_test)), 2) AS
proportion
FROM employee_test
GROUP BY gender;
```

```
-- 4. Group Employee age into 5 categories (20 – 29, 30 – 39, 40-49, 50-59, >60). What age
group has the highest and lowest employee?
SELECT
  CASE
    WHEN age BETWEEN 20 AND 29 THEN '20-29'
    WHEN age BETWEEN 30 AND 39 THEN '30-39'
    WHEN age BETWEEN 40 AND 49 THEN '40-49'
    WHEN age BETWEEN 50 AND 59 THEN '50-59'
    ELSE '>60'
  END AS AgeGroup,
  COUNT(employee_id) AS EmployeeCount
FROM employee_test
GROUP BY AgeGroup
ORDER BY EmployeeCount DESC;


-- 5. Who works in the Finance department?
SELECT *
FROM employee_test
WHERE department LIKE '%Finance%';

-- 6. Who has the highest average training score among all employees?
SELECT employee_id, ROUND(AVG(avg_training_score),1) AS AvgTrainingScore
FROM employee_test
GROUP BY employee_id
ORDER BY AvgTrainingScore DESC
LIMIT 1;

-- 7. Which regions have the highest number of departures (employees who have left), and
what are the corresponding departments?
SELECT et.region, et.department, COUNT(et.employee_id) AS Departures
FROM employee_perf AS ep
RIGHT JOIN employee_test AS et
      ON ep.employee_id = et.employee_id
WHERE ep.Attrition = 'Yes'
GROUP BY et.region, et.department
ORDER BY Departures DESC;

-- 8. Which department has the most employees, and which department has the fewest
employees?
SELECT department, COUNT(employee_id) AS EmployeeCount
```

```sql
FROM employee_test
GROUP BY department
ORDER BY EmployeeCount DESC;



-- 9. Who are the top 5 highest-earning employees in the 'Technology' department?

SELECT ep.*
FROM employee_perf AS ep
INNER JOIN employee_test AS et
        ON ep.employee_id = et.employee_id
WHERE et.department = 'Technology'
ORDER BY ep.MonthlyIncome DESC
LIMIT 5;

-- 10.        Who are the employees with awards in departments with more than 100
employees, and what are their department names?

SELECT ep.*, et.department
FROM employee_perf AS ep
INNER JOIN employee_test AS et
        ON ep.employee_id = et.employee_id
WHERE et.awards_won = 1
AND et.department IN (
   SELECT department
   FROM employee_test
   GROUP BY department
   HAVING COUNT(employee_id) > 100
);




-- Which department has the most employees, and which department has the fewest
employees?
SELECT department, COUNT(employee_id) AS EmployeeCount
FROM employee_test
GROUP BY department
ORDER BY EmployeeCount DESC;


-- Who are the top 5 highest-earning employees in the 'Technology' department?
SELECT ep.*
FROM employee_perf AS ep
INNER JOIN employee_test AS et
      ON ep.employee_id = et.employee_id
WHERE et.department = 'Technology'
```

```
ORDER BY ep.MonthlyIncome DESC
LIMIT 5;


-- Who are the employees with awards in departments with more than 100 employees, and what are their department names?
SELECT ep.*, et.department
FROM employee_perf AS ep
INNER JOIN employee_test AS et
        ON ep.employee_id = et.employee_id
WHERE et.awards_won = 1
AND et.department IN (
   SELECT department
   FROM employee_test
   GROUP BY department
   HAVING COUNT(employee_id) > 100
);


-- What is the average training score of employees in each department
SELECT department,  ROUND(AVG(avg_training_score),0) AS AvgTrainingScore
FROM employee_test
GROUP BY department;


-- What is the average previous year rating by department?
SELECT department, ROUND(AVG(previous_year_rating),0) AS AvgPrevYearRating
FROM employee_test
GROUP BY department;


-- What is the average training score of employees by education type?
SELECT education, ROUND(AVG(avg_training_score),0) AS AvgTrainingScore
FROM employee_test
GROUP BY education;


-- Group Average training score into grades (A,B,C,D,E,F) and what grade had the highest and lowest number of employees
SELECT
   CASE
      WHEN avg_training_score >= 90 THEN 'A'
      WHEN avg_training_score >= 80 THEN 'B'
      WHEN avg_training_score >= 70 THEN 'C'
      WHEN avg_training_score >= 60 THEN 'D'
      WHEN avg_training_score >= 50 THEN 'E'
      ELSE 'F'
   END AS TrainingGrade,
   COUNT(employee_id) AS EmployeeCount
FROM employee_test
GROUP BY TrainingGrade
ORDER BY EmployeeCount DESC;


-- Which three departments have the highest average job satisfaction among employees with a
Bachelor's degree?
SELECT et.department, ROUND(AVG(ep.JobSatisfaction),0) AS AvgJobSatisfaction
FROM employee_perf AS ep
INNER JOIN employee_test AS et
        ON ep.employee_id = et.employee_id
WHERE et.education = 'Bachelor''s'
GROUP BY et.department
ORDER BY AvgJobSatisfaction DESC
LIMIT 3;


-- What is the average previous year rating by recruitment channel?
SELECT recruitment_channel, ROUND(AVG(previous_year_rating),0) AS
AvgPrevYearRating
FROM employee_test
GROUP BY recruitment_channel;


-- What is the split of gender by the previous year rating?

SELECT
   previous_year_rating,
   gender,
   COUNT(*) AS gender_count
   ROUND((COUNT(*) * 100.0 / (SELECT COUNT(*) FROM employee_test)), 2) AS
proportion
FROM
   employee_test
GROUP BY
   previous_year_rating, gender
ORDER BY
   previous_year_rating, gender;


-- Based on the age group created what is the average previous year rating and average
training score.
SELECT AgeGroup, ROUND(AVG(previous_year_rating),0) AS AvgPrevYearRating,
ROUND(AVG(avg_training_score),0) AS AvgTrainingScore
FROM (
   SELECT
```

```
        CASE
            WHEN age BETWEEN 20 AND 29 THEN '20-29'
            WHEN age BETWEEN 30 AND 39 THEN '30-39'
            WHEN age BETWEEN 40 AND 49 THEN '40-49'
            WHEN age BETWEEN 50 AND 59 THEN '50-59'
            ELSE '>60'
        END AS AgeGroup,
        previous_year_rating, avg_training_score
    FROM employee_test
) AS AgeGroupedData
GROUP BY AgeGroup;


-- What is the average age of male and female employees, and how many employees are there
for each gender?
SELECT gender, ROUND(AVG(age)) AS AvgAge, COUNT(employee_id) AS
EmployeeCount
FROM employee_test
GROUP BY gender;

-- Who are the top 5 highest-earning employees with a JobLevel of 3 or higher?
SELECT employee_id, MonthlyIncome, JobLevel
FROM employee_perf
WHERE JobLevel >= 3
ORDER BY MonthlyIncome DESC
LIMIT 5;
```