DATA ANALYST PORTFOLIO SQL PROJECT - HR Data

TESTING TABLEAU/ POWER BI REPORTS IN SQL

😍

Michelle King

www.MichelleMolneyKing.com

Introduction

This code is a set of SQL queries and commands for creating a table named "hrdata" and importing data from a CSV file into the table. The "hrdata" table contains various fields such as employee number, gender, age, department, education, job role, attrition, etc.

The queries then perform various analyses on the data such as calculating employee count, attrition count and rate, active employee count, average age, attrition by gender, department-wise attrition, education field-wise attrition, and job satisfaction rating. These analyses provide insights into the HR data and can be used to make informed decisions related to employee management, retention, and recruitment.

This code demonstrates how SQL queries and commands can be used to create a table, import data into the table, and perform various analyses on the data to gain insights and make informed decisions.

SQL Aims

1. Create a table named "hrdata" with various fields to store employee information.
2. Import data from a CSV file into the "hrdata" table.
3. Analyze the data in the "hrdata" table to gain insights into the workforce.
4. Calculate the total number of employees in the company.
5. Count the number of employees who have left the company (i.e., attrition = 'Yes').
6. Calculate the attrition rate by dividing the number of employees who have left the company by the total number of employees.
7. Calculate the number of active employees in the company.
8. Calculate the average age of employees in the company.
9. Count the number of employees who have left the company by gender.
10. Count the number of employees who have left the company by department and calculate the percentage of employees who have left in each department.
11. Group employees by age and count the number of employees in each age group.
12. Count the number of employees who have left the company by education field.
13. Count the number of employees who have left the company by age group and gender and calculate the percentage of employees who have left in each group.
14. Create a cross-tabulation of job roles and job satisfaction ratings, showing the number of employees in each combination.
15. The aims of these SQL queries and commands are to provide insights into the HR data and help businesses make informed decisions related to employee management, retention, and recruitment.

Here's some more information on the different SQL queries and commands used in the code:

Creating a Table

The code starts by creating a table, HR data, named "hrdata" using the "CREATE TABLE" SQL command. The table contains various fields such as employee number, gender, age, department, education, job role, attrition, etc. The "PRIMARY KEY" constraint is applied to the "emp_no" field to ensure that each employee has a unique identifier.

Importing Data into the Table

The code then uses the "COPY" SQL command to import data from a CSV file into the "hrdata" table. The file path and delimiter are specified in the command.

Analysing the Data

The code then performs various SQL queries to analyze the data in the "hrdata" table. Here's a brief summary of each query:

- Employee Count: The "SUM" SQL function is used to calculate the total number of employees in the "hrdata" table by summing the "employee_count" field.

Analysing the Data

- Attrition Count: The "COUNT" SQL function is used to count the number of employees who have left the company (i.e., attrition = 'Yes').
- Attrition Rate: The attrition rate is calculated by dividing the number of employees who have left the company by the total number of employees and multiplying by 100. The "ROUND" SQL function is used to round the result to two decimal places.
- Active Employee: The number of active employees is calculated by subtracting the number of employees who have left the company from the total number of employees.
- Average Age: The "AVG" SQL function is used to calculate the average age of employees in the "hrdata" table.
- Attrition by Gender: The "COUNT" and "GROUP BY" SQL functions are used to count the number of employees who have left the company by gender.

- Department-wise Attrition: The "COUNT", "GROUP BY", and "ORDER BY" SQL functions are used to count the number of employees who have left the company by department and calculate the percentage of employees who have left in each department.
- No of Employee by Age Group: The "GROUP BY" and "ORDER BY" SQL functions are used to group employees by age and count the number of employees in each age group.
- Education Field-wise Attrition: The "COUNT", "GROUP BY", and "ORDER BY" SQL functions are used to count the number of employees who have left the company by education field.
- Attrition Rate by Gender for different Age Groups: The "COUNT", "GROUP BY", and "ORDER BY" SQL functions are used to count the number of employees who have left the company by age group and gender and calculate the percentage of employees who have left in each group.
- Job Satisfaction Rating: The "CREATE EXTENSION" and "CROSSTAB" SQL commands are used to create a cross-tabulation of job roles and job satisfaction ratings, showing the number of employees in each combination.

These queries provide insights into the HR data and can be used to make informed decisions related to employee management, retention, and recruitment.

Businesses can use the insights gained from these queries to improve their workforce in several ways. Here are a few examples:

- Retention Strategies: By analyzing attrition rates and patterns, businesses can identify the root causes of employee turnover, such as dissatisfaction with work, poor management, or lack of growth opportunities. Based on these insights, businesses can implement retention strategies such as improving employee engagement, providing training and development opportunities, and offering competitive compensation and benefits.
- Recruitment Strategies: The queries can also help identify the departments, job roles, and education fields with the highest attrition rates, which can inform recruitment strategies. For example, if a particular department or job role has a high attrition rate, businesses can focus on recruiting candidates with the skills and experience necessary to succeed in that role, as well as providing a supportive work environment.
- Employee Engagement: The queries can help identify job roles or departments with lower job satisfaction ratings. By addressing the factors that lead to lower job satisfaction, such as lack of recognition or opportunities for growth, businesses can increase employee engagement and motivation, leading to higher productivity and better performance.
- Workforce Planning: The queries can help businesses understand the demographics and characteristics of their workforce, such as age, gender, education, and job roles. This information can inform workforce planning by identifying areas with skills gaps or imbalances, allowing businesses to recruit or train employees with the necessary skills and experience.

The insights gained from these queries can provide businesses with a deeper understanding of their workforce and help them make informed decisions about recruitment, retention, and employee engagement. By taking action based on these insights, businesses can improve their workforce and achieve their objectives.

SQL

Create Table

```
create table hrdata
(
	emp_no int8 PRIMARY KEY,
	gender varchar(50) NOT NULL,
	marital_status varchar(50),
	age_band varchar(50),
	age int8,
	department varchar(50),
	education varchar(50),
	education_field varchar(50),
	job_role varchar(50),
	business_travel varchar(50),
	employee_count int8,
	attrition varchar(50),
	attrition_label varchar(50),
	job_satisfaction int8,
	active_employee int8
)
```

Import Data in Table Using Query

```
COPY hrdata FROM 'D:\hrdata.csv' DELIMITER ',' CSV HEADER;
```

Employee Count:

```
select sum(employee_count) as Employee_Count from hrdata;
```

Attrition Count:

```
select count(attrition) from hrdata where attrition='Yes';
```

Attrition Rate:

```
select
round (((select count(attrition) from hrdata where attrition='Yes')/
sum(employee_count)) * 100,2)
from hrdata;
```

Active Employee:

```
select sum(employee_count) - (select count(attrition) from hrdata  where
attrition='Yes') from hrdata;
```

OR

```
select (select sum(employee_count) from hrdata) - count(attrition) as
active_employee from hrdata
where attrition='Yes';
```

Average Age:

```
select round(avg(age),0) from hrdata;
```

Attrition by Gender

```
select gender, count(attrition) as attrition_count from hrdata
where attrition='Yes'
group by gender
order by count(attrition) desc;
```

Department wise Attrition:

```
select department, count(attrition), round((cast (count(attrition) as numeric) /
(select count(attrition) from hrdata where attrition= 'Yes')) * 100, 2) as pct from
hrdata
where attrition='Yes'
group by department
```

```
order by count(attrition) desc;
```

No of Employee by Age Group

```
SELECT age,  sum(employee_count) AS employee_count FROM hrdata
GROUP BY age
order by age;
```

Education Field wise Attrition:

```
select education_field, count(attrition) as attrition_count from hrdata
where attrition='Yes'
group by education_field
order by count(attrition) desc;
```

Attrition Rate by Gender for different Age Group

```
select age_band, gender, count(attrition) as attrition,
round((cast(count(attrition) as numeric) / (select count(attrition) from hrdata
where attrition = 'Yes')) * 100,2) as pct
from hrdata
where attrition = 'Yes'
group by age_band, gender
order by age_band, gender desc;
```

Job Satisfaction Rating

```
-Run this query first to activate the cosstab() function in postgres
CREATE EXTENSION IF NOT EXISTS tablefunc;

-Then run this to get o/p-
SELECT *
FROM crosstab(
 'SELECT job_role, job_satisfaction, sum(employee_count)
  FROM hrdata
  GROUP BY job_role, job_satisfaction
  ORDER BY job_role, job_satisfaction'
          ) AS ct(job_role varchar(50), one numeric, two numeric, three numeric, four numeric)
ORDER BY job_role;
```