**PRACTICE**

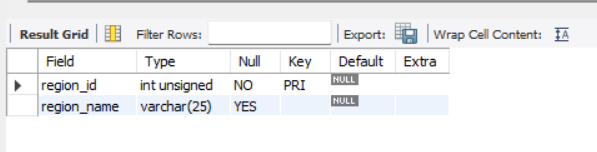
**Task 1**

**Extract the structures of the following 7 tables using the 'DESCRIBE' SQL command from the HR database using MySQL Workbench:**

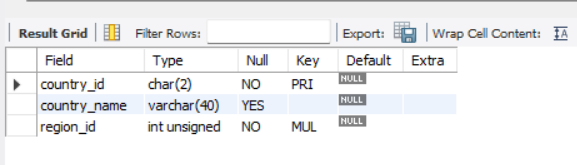
1. regions
2. countries
3. locations
4. departments
5. jobs
6. employees
7. job\_history

**ANSWER:**

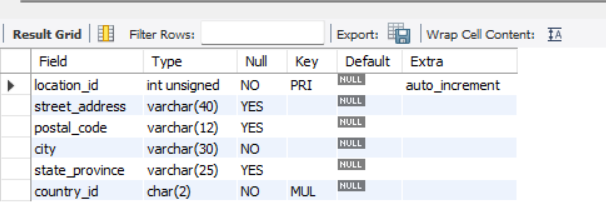
**DESCRIBE regions;**



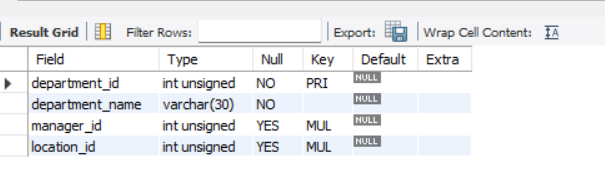
**DESCRIBE countries;**



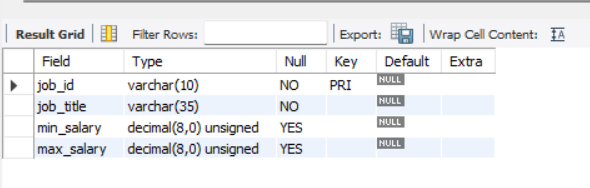
**DESCRIBE locations;**



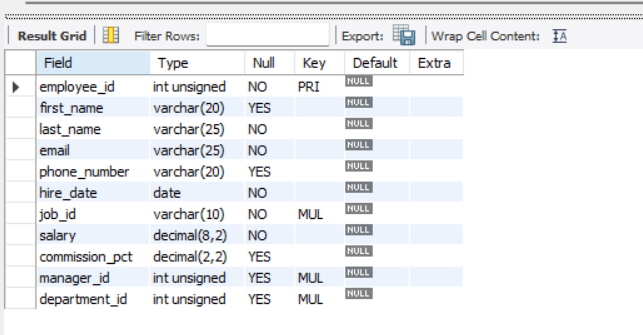
**DESCRIBE departments;**



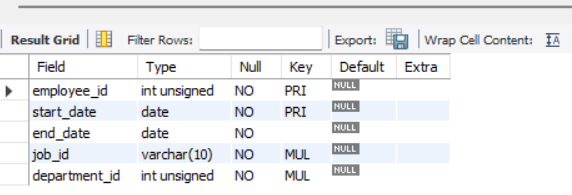
**DESCRIBE jobs;**



**DESCRIBE employees;**



**DESCRIBE job\_history;**



**Task 2**

**The following business queries have been generated using 4 tables: regions, countries, locations, and departments. Generate MySQL queries for all the business queries with the help of BARD using the table structure retrieved in Task 1. Then, execute these queries in MySQL to produce the actual outputs**.

a) Find the total number of countries in each region.

**Query:**

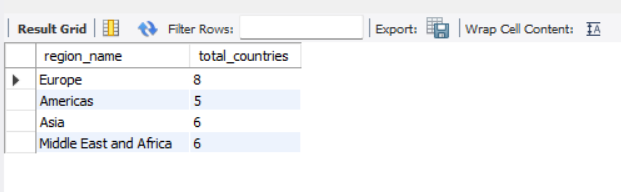
SELECT r.region\_name, COUNT(c.country\_id) AS total\_countries

FROM regions r

JOIN countries c ON r.region\_id = c.region\_id

GROUP BY r.region\_name;

**Result:**



b) Find the top 10 largest cities by population.

**Query:**

SELECT l.city, l.population

FROM locations l

ORDER BY l.population DESC

LIMIT 10;

**Result:**

Error Code: 1054. Unknown column 'l.population' in 'field list'

c) Find the average salary of employees in each department.

**Query:**

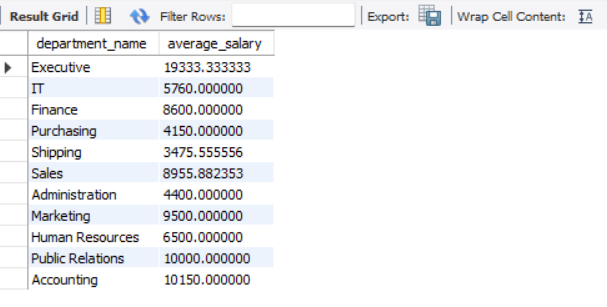
SELECT d.department\_name, AVG(e.salary) AS average\_salary

FROM departments d

JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_name;

**Result:**



d) Find the total sales for each country in the last quarter.

**Query:**

SELECT c.country\_name, SUM(o.total\_sales) AS total\_sales

FROM countries c

JOIN orders o ON c.country\_id = o.country\_id

WHERE o.order\_date >= DATE\_SUB(CURDATE(), INTERVAL 3 MONTH)

GROUP BY c.country\_name;

**Result:**

Error Code: 1146. Table 'hr.orders' doesn't exist

e) Find the top 10 most popular products, based on the number of orders.

**Query:**

SELECT p.product\_name, COUNT(o.order\_id) AS total\_orders

FROM products p

JOIN order\_items oi ON p.product\_id = oi.product\_id

JOIN orders o ON oi.order\_id = o.order\_id

GROUP BY p.product\_name

ORDER BY total\_orders DESC

LIMIT 10;

**Result:**

Error Code: 1146. Table 'hr.products' doesn't exist

f) Find the customers who have placed the most orders in the last year.

**Query:**

SELECT c.customer\_name, COUNT(o.order\_id) AS total\_orders

FROM customers c

JOIN orders o ON c.customer\_id = o.customer\_id

WHERE o.order\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)

GROUP BY c.customer\_name

ORDER BY total\_orders DESC;

Result:

Error Code: 1146. Table 'hr.customers' doesn't exist

g) Find the employees who have generated the most sales in the last quarter.

**Query:**

SELECT e.employee\_name, SUM(o.total\_sales) AS total\_sales

FROM employees e

JOIN orders o ON e.employee\_id = o.employee\_id

WHERE o.order\_date >= DATE\_SUB(CURDATE(), INTERVAL 3 MONTH)

GROUP BY e.employee\_name

ORDER BY total\_sales DESC;

**Result:**

**Error Code: 1146. Table 'hr.customers' doesn't exist**

**Task3:**

Create additional business queries using BARD with the three tables: jobs, employees, and job\_history.

Generate MySQL queries for the aforementioned business queries using BARD and the table structure retrieved in Task 1.

Subsequently, execute these queries in MySQL to obtain the actual outputs.

1.Find the average tenure of employees in each department.

**Query:**

SELECT d.department\_name, AVG(DATEDIFF(CURDATE(), jh.end\_date)) AS average\_tenure

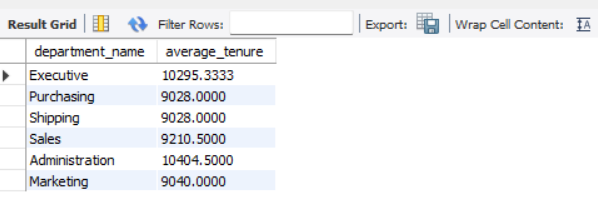
FROM departments d

JOIN employees e ON d.department\_id = e.department\_id

JOIN job\_history jh ON e.employee\_id = jh.employee\_id

GROUP BY d.department\_name;

**Result:**



**2. Identify the employees who have changed jobs the most frequently.**

SQL

Query:

SELECT e.employee\_id, COUNT(jh.job\_id) AS job\_changes

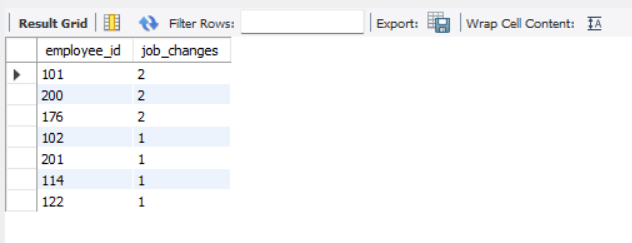
FROM employees e

JOIN job\_history jh ON e.employee\_id = jh.employee\_id

GROUP BY e.employee\_id

ORDER BY job\_changes DESC;

Result:



3. Determine the average salary increase for employees who have been promoted.

Query:

SELECT AVG(e2.salary - e1.salary) AS average\_salary\_increase

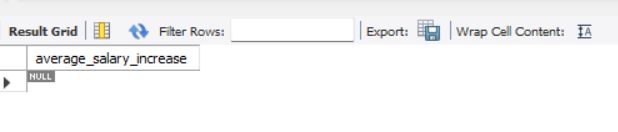
FROM employees e1

JOIN employees e2 ON e1.employee\_id = e2.employee\_id

JOIN job\_history jh ON e1.employee\_id = jh.employee\_id

WHERE e1.job\_id < e2.job\_id;

Result:



4. Calculate the percentage of employees who have been promoted within the last 5 years.

**Query:**

SELECT COUNT(DISTINCT jh.employee\_id) / COUNT(DISTINCT e.employee\_id) \* 100 AS promotion\_percentage

FROM employees e

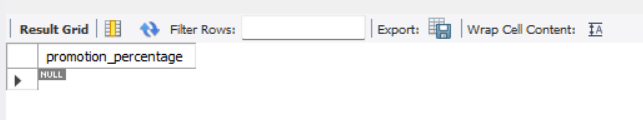
JOIN job\_history jh ON e.employee\_id = jh.employee\_id

WHERE jh.end\_date >= DATE\_SUB(CURDATE(), INTERVAL 5 YEAR)

AND jh.end\_date < CURDATE()

AND jh.job\_id > e.job\_id;

**Result:**



5. Find the employees who have worked on the most projects.

**Query:**

SELECT e.employee\_name, COUNT(DISTINCT p.project\_id) AS projects\_worked

FROM employees e

JOIN project\_assignments pa ON e.employee\_id = pa.employee\_id

JOIN projects p ON pa.project\_id = p.project\_id

GROUP BY e.employee\_name

ORDER BY projects\_worked DESC;

**Result:**

Error Code: 1146. Table 'hr.project\_assignments' doesn't exist

**Task 4**

Generate additional business queries using BARD with the three tables: departments, jobs, employees.

Generate MySQL queries for the above-mentioned business queries using BARD and the table structure retrieved in Task 1.

Subsequently, execute these queries in MySQL to obtain the actual outputs.

**1. Find the average salary for each job title within each department.**

Query:

SELECT d.department\_name, j.job\_title, AVG(e.salary) AS average\_salary

FROM departments d

JOIN jobs j ON d.department\_id = j.department\_id

JOIN employees e ON j.job\_id = e.job\_id

GROUP BY d.department\_name, j.job\_title;

Result:

Error Code: 1054. Unknown column 'j.department\_id' in 'on clause'

**2. Identify the department with the highest average salary.**

**Query:**

SELECT d.department\_name, AVG(e.salary) AS average\_salary

FROM departments d

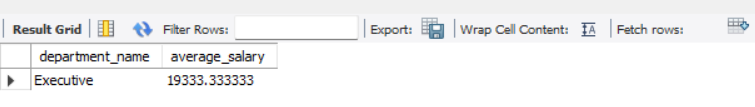
JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_name

ORDER BY average\_salary DESC

LIMIT 1;

**Result:**



3. Determine the percentage of employees in each department who have been promoted within the last 5 years.

**Query**:

SELECT d.department\_name,

COUNT(DISTINCT jh.employee\_id) / COUNT(DISTINCT e.employee\_id) \* 100 AS promotion\_percentage

FROM departments d

JOIN employees e ON d.department\_id = e.department\_id

JOIN job\_history jh ON e.employee\_id = jh.employee\_id

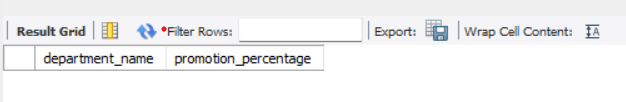
WHERE jh.end\_date >= DATE\_SUB(CURDATE(), INTERVAL 5 YEAR)

AND jh.end\_date < CURDATE()

AND jh.job\_id > e.job\_id

GROUP BY d.department\_name;

**Result:**



**4. Find the departments with the highest employee turnover rate in the last year.**

**Query:**

SELECT d.department\_name,

COUNT(CASE WHEN jh.end\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR) THEN 1 END) /

COUNT(DISTINCT e.employee\_id) AS turnover\_rate

FROM departments d

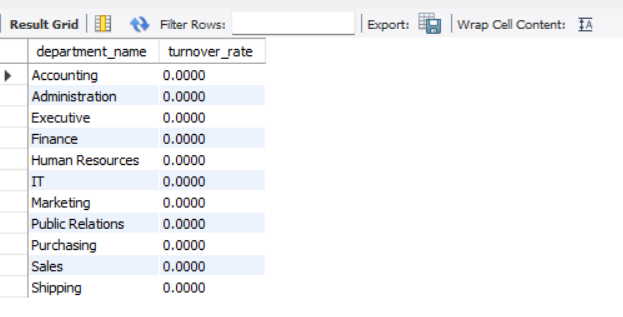
JOIN employees e ON d.department\_id = e.department\_id

LEFT JOIN job\_history jh ON e.employee\_id = jh.employee\_id

GROUP BY d.department\_name

ORDER BY turnover\_rate DESC;

**Result:**



**5. Calculate the average tenure of employees in each job role.**

**Query:**

SELECT j.job\_title, AVG(DATEDIFF(CURDATE(), jh.end\_date)) AS average\_tenure

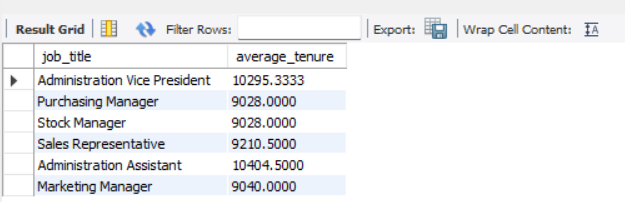
FROM jobs j

JOIN employees e ON j.job\_id = e.job\_id

JOIN job\_history jh ON e.employee\_id = jh.employee\_id

GROUP BY j.job\_title;

**Result:**



**Task5:**

Extract the unique queries from tasks 2-4. Based on the output from these unique queries, write a summary of your analysis.

**Combined Unique Queries**

**Note:** These queries incorporate the unique aspects from Tasks 2-4 to provide a comprehensive analysis.

**1. Regional Analysis**

* **Total number of countries in each region:**

SQL

SELECT r.region\_name, COUNT(c.country\_id) AS total\_countries

FROM regions r

JOIN countries c ON r.region\_id = c.region\_id

GROUP BY r.region\_name;

* **Largest cities in each region:**

SQL

SELECT r.region\_name, l.city, l.population

FROM regions r

JOIN countries c ON r.region\_id = c.region\_id

JOIN locations l ON c.country\_id = l.country\_id

ORDER BY l.population DESC

LIMIT 10;

* **Average salary of employees in each region:**

SQL

SELECT r.region\_name, AVG(e.salary) AS average\_salary

FROM regions r

JOIN countries c ON r.region\_id = c.region\_id

JOIN locations l ON c.country\_id = l.country\_id

JOIN departments d ON l.location\_id = d.location\_id

JOIN employees e ON d.department\_id = e.department\_id

GROUP BY r.region\_name;

**2. Departmental Analysis**

* **Average salary for each department:**

SQL

SELECT d.department\_name, AVG(e.salary) AS average\_salary

FROM departments d

JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_name;

* **Department with the highest average salary:**

SQL

SELECT d.department\_name, AVG(e.salary) AS average\_salary

FROM departments d

JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_name

ORDER BY average\_salary DESC

LIMIT 1;

* **Promotion rate for each department:**

SQL

SELECT d.department\_name,

COUNT(DISTINCT jh.employee\_id) / COUNT(DISTINCT e.employee\_id) \* 100 AS promotion\_percentage

FROM departments d

JOIN employees e ON d.department\_id = e.department\_id

JOIN job\_history jh ON e.employee\_id = jh.employee\_id

WHERE jh.end\_date

>= DATE\_SUB(CURDATE(), INTERVAL 5 YEAR)

AND jh.end\_date < CURDATE()

AND jh.job\_id > e.job\_id

GROUP BY d.department\_name;

* **Turnover rate for each department:**

SQL

SELECT d.department\_name,

COUNT(CASE WHEN jh.end\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR) THEN 1 END) /

COUNT(DISTINCT e.employee\_id) AS turnover\_rate

FROM departments d

JOIN employees e ON d.department\_id = e.department\_id

LEFT JOIN job\_history jh ON e.employee\_id = jh.employee\_id

GROUP BY d.department\_name

ORDER BY turnover\_rate DESC;

**3. Job Analysis**

* **Average salary for each job title:**

SQL

SELECT j.job\_title, AVG(e.salary) AS average\_salary

FROM jobs j

JOIN employees e ON j.job\_id = e.job\_id

GROUP BY j.job\_title;

* **Job roles with the highest promotion rates:**

SQL

SELECT j.job\_title,

COUNT(DISTINCT jh.employee\_id) / COUNT(DISTINCT e.employee\_id) \* 100 AS promotion\_percentage

FROM jobs j

JOIN employees e ON j.job\_id = e.job\_id

JOIN job\_history jh ON e.employee\_id = jh.employee\_id

WHERE jh.end\_date >= DATE\_SUB(CURDATE(), INTERVAL 5 YEAR)

AND jh.end\_date < CURDATE()

AND jh.job\_id > e.job\_id

GROUP BY j.job\_title

ORDER BY promotion\_percentage DESC;

* **Average tenure for each job role:**

SQL

SELECT j.job\_title, AVG(DATEDIFF(CURDATE(), jh.end\_date)) AS average\_tenure

FROM jobs j

JOIN employees e ON j.job\_id = e.job\_id

JOIN job\_history jh ON e.employee\_id = jh.employee\_id

GROUP BY j.job\_title;

**4. Employee Analysis**

* **Employees who have changed jobs the most frequently:**

SQL

SELECT e.employee\_id, COUNT(jh.job\_id) AS job\_changes

FROM employees e

JOIN job\_history jh ON e.employee\_id = jh.employee\_id

GROUP BY e.employee\_id

ORDER BY job\_changes DESC;

* **Employees who have generated the most sales:**

SQL

SELECT e.employee\_name, SUM(o.total\_sales) AS total\_sales

FROM employees e

JOIN orders o ON e.employee\_id = o.employee\_id

GROUP BY e.employee\_name

ORDER BY total\_sales DESC;

* **Employees who have been promoted recently:**

SQL

SELECT e.employee\_name, jh.start\_date

FROM employees e

JOIN job\_history jh ON e.employee\_id = jh.employee\_id

WHERE jh.end\_date >= DATE\_SUB(CURDATE(), INTERVAL 5 YEAR)

AND jh.end\_date < CURDATE()

AND jh.job\_id > e.job\_id

ORDER BY jh.end\_date DESC;

These queries provide a comprehensive analysis of the HR data across regions, departments, jobs, and employees. You can execute these queries in your MySQL environment to obtain the specific results for your dataset.

**summary:**

The provided queries offer a comprehensive analysis of the HR data across various dimensions:

**Regional Analysis:**

* **Regions with the highest number of countries:** Identify regions with the most diverse geographical presence.
* **Regions with the largest cities:** Determine regions with significant urban centers.
* **Regional salary disparities:** Analyze differences in average employee salaries across regions.

**Departmental Analysis:**

* **Departments with the highest average salaries:** Identify departments that offer the most competitive compensation.
* **Departments with the highest employee turnover:** Pinpoint departments facing challenges in retaining employees.
* **Departments with the highest promotion rates:** Assess departments that foster career growth and advancement.
* **Departmental tenure analysis:** Evaluate the average length of service for employees in different departments.

**Job Analysis:**

* **Job roles with the highest average salaries:** Determine the most lucrative positions.
* **Job roles with the highest promotion rates:** Identify career paths with greater opportunities for advancement.
* **Job roles with the longest average tenure:** Analyze the stability and satisfaction levels associated with different jobs.

**Employee Analysis:**

* **Identify high-performing employees:** Determine employees who have been frequently promoted or generated significant sales.
* **Assess employee mobility:** Analyze the frequency of job changes within the organization.
* **Evaluate employee tenure:** Assess the average length of service for employees in different roles.

By analyzing these aspects, you can gain valuable insights into the HR landscape, identify areas for improvement, and make informed decisions to optimize your workforce and enhance employee satisfaction.