

# **SQL-Mongo Project – IBM HR Analytics Employee Attrition & Performance**

**BUAN 6320**

**GROUP 8**

**Venkat Sai Kiran Bachu (VXB220061)**

**Nikhil Inturi (NXI220006)**

**Shiva Kumar Reddy Koppula (S XK230064)**

**Rupa Reddy Nandikonda (RXR230021)**

**Varnika Reddy Babireddy (VXB230010)**

**Akshara Reddy Mutyala (ARM230002)**

# **Contents**

- 1. Relational Data Model**
- 2. Assumptions about Data Entities and Relationships**
- 3. Assumptions about Dataset**
- 4. SQL Queries**
- 5. Mongodb Queries**

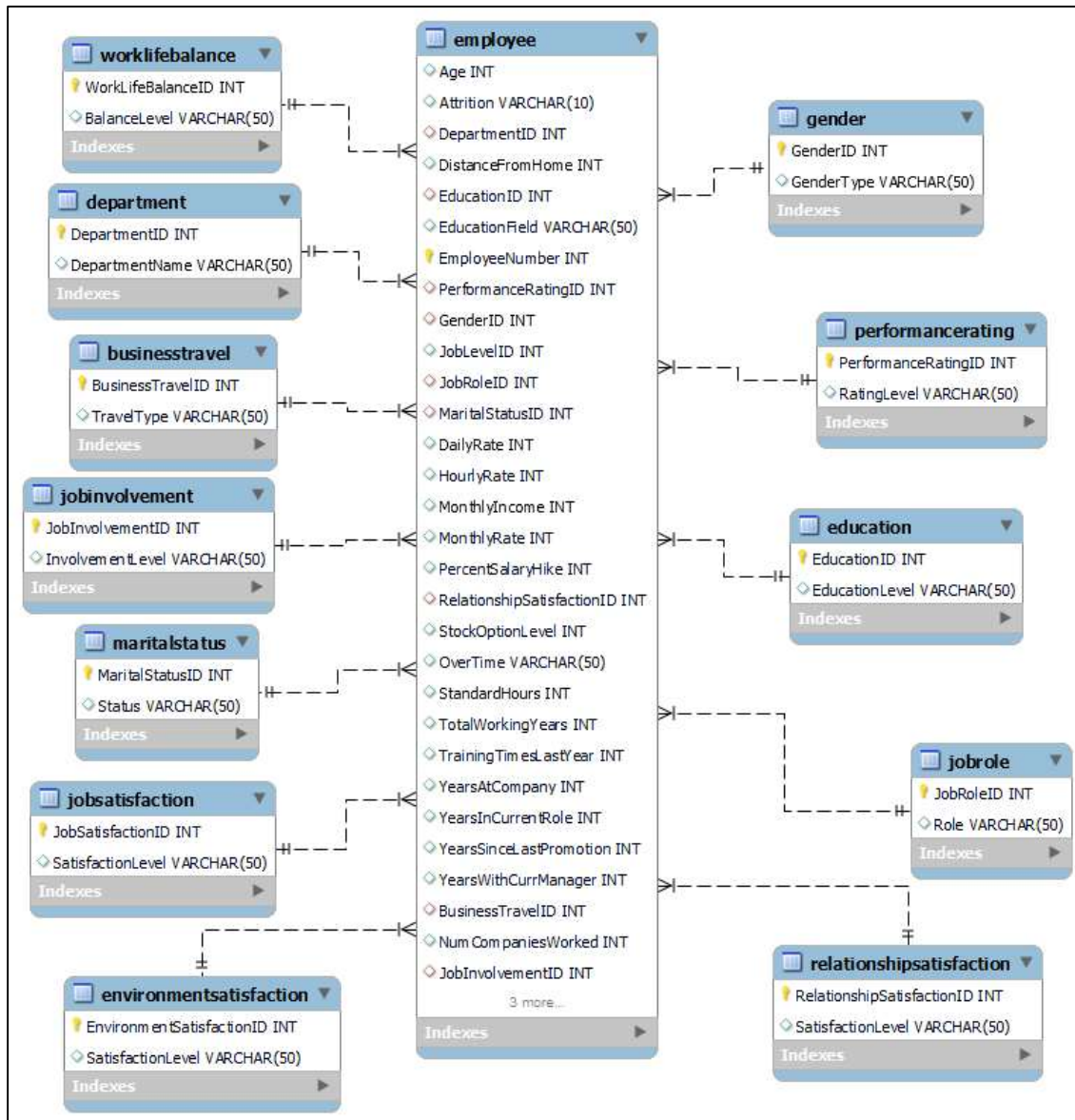
## **Relational Data Model**

### **Assumptions/Notes About Data Entities and Relationships**

The following 20 database design assumptions are derived from the available data:

- 1.** Employee Uniqueness: EmployeeNumber serves as a unique means of identifying each employee.
- 2.** Attrition Values: The EmployeeDetails table's Attrition field can only have the values "Yes" or "No"
- 3.** Business Travel Categories: Only some categories, such "Travel\_Rarely," "Travel Frequently," or "Non-Travel," are included in the EmployeeDetails table's BusinessTravel column.
- 4.** Education Levels: Employees' educational attainment is indicated in the EmployeeDetails table's Education column, which is restricted to the specified categories.
- 5.** EnvironmentSatisfactionLevels: The EmployeeDetails table's EnvironmentSatisfaction column only includes the categories that are given and relates to the degree of satisfaction.
- 6.** Gender Values: Within the EmployeeDetails database, the Gender field can only contain the values 'Male' or 'Female'.
- 7.** Job Involvement Levels: The JobInvolvement column in the EmployeeDetails table corresponds to involvement levels and is limited to the provided categories.
- 8.** Job Roles: The JobRole column in the EmployeeDetails table corresponds to specific job roles and is limited to the provided categories.
- 9.** Marital Status Values: The EmployeeDetails table's MaritalStatus column can only have the values "Single," "Married," or "Divorced."
- 10.** Performance Ratings: Performance ratings are reflected in the EmployeeDetails table's PerformanceRating column, which is restricted to the specified categories.
- 11.** Relationship Satisfaction Levels: The RelationshipSatisfaction column in the EmployeeDetails table corresponds to satisfaction levels and is limited to the provided categories.
- 12.** Work-Life Balance Levels: The WorkLifeBalance column in the EmployeeDetails table corresponds to balance levels and is limited to the provided categories.
- 13.** Department Uniqueness: Each department in the Department table is uniquely identified by its DepartmentID.
- 14.** Education Levels Uniqueness: Each education level in the Education table is uniquely identified by its EducationID.
- 15.** Job Roles Uniqueness: Each job role in the JobRole table is uniquely identified by its JobRoleID.
- 16.** No Null Values: All required fields in the database tables are presumed to have non- null values, particularly those associated with foreign key

## Entity-Relationship Diagram



In the phase 1 document, we had separated the salary related data into “salary” table, experience related data into “experience” table and job review data in “jobreviewdetails” table because of this employee number will be 3 times additionally loaded into the database and this occupies much space in database we created. As we realized this, we added all the columns and data related to salary, experience and job review into one single table “employee”. By, this data is also in 3NF form.

## Original Database

### Assumptions/Notes About Data Set

#### Revised Table Structure:

1. We used Label Encoding for the dataset to convert the string values (VARCHAR) to numeric (INT). After normalization, these numeric values are used as Primary Keys in the Lookup Tables. For the 'Department' table, we have encoded the three categories: 'Sales,' 'Research & Development,' and 'Human Resources' into 1, 2, and 3, respectively.
2. For the 'Education' table, we have encoded the five categories: 'Below College', 'College', 'Bachelor', 'Master', and 'Doctor' into 1, 2, 3, 4, and 5, respectively.
3. For the 'EnvironmentSatisfaction' table, we have encoded the four categories: 'Low', 'Medium', 'High', and 'Very High' into 1, 2, 3, and 4, respectively.
4. For the 'gender' table, we have encoded the categories 'Male' and 'Female' into M and F, respectively.
5. For the 'JobInvolvement' table, we have encoded the four categories: 'Low', 'Medium', 'High', and 'Very High' into 1, 2, 3, and 4, respectively.
6. For the 'JobLevel' table, we encoded five categories: 'Entry Level', 'Intermediate', 'Senior', 'Manager', and 'Executive' as 1, 2, 3, 4, and 5.
7. We have encoded the three categories in the 'MaritalStatus' table: 'Single' is encoded as 1, 'Married' as 2, and 'Divorced' as 3.
8. We have encoded the categories 'Low', 'Good', 'Excellent', and 'Outstanding' into 1, 2, 3, and 4 for the 'PerformanceRating' table.
9. For the 'RelationshipSatisfaction' table, we have coded the categories 'Low', 'Medium', 'High', and 'Very High' as 1, 2, 3, and 4, respectively.
10. For the 'WorkLifeBalance' table, we have mapped 'Bad', 'Good', 'Better', and 'Best' to 1, 2, 3, and 4.
11. For the 'JobSatisfaction' table, we have coded the categories 'Low', 'Medium', 'High' and 'Very High' as 1,2,3,4 respectively
12. For the 'BusinessTravel' table, we have coded the categories 'Non-Travel', 'TravelRarely', and 'TravelFrequently' as 1,2,3 respectively

## ***Data in the Database***

### **Education**

EducationID

EducationLevel

### **EnvironmentSatisfaction**

EnvironmentSatisfactionID

SatisfactionLevel

### **JobInvolvement**

JobInvolvementID

InvolvementLevel

### **JobRole**

JobRoleID

Role

### **JobSatisfaction**

JobSatisfactionID

SatisfactionLevel

### **MaritalStatus**

MaritalStatusID

Status

### **BusinessTravel**

BusinessTravelID

TravelType

### **Gender**

GenderID

GenderType

### **PerformanceRating**

PerformanceRatingID

RatingLevel

### **RelationshipSatisfaction**

RelationshipSatisfactionID

SatisfactionLevel

### **WorkLifeBalance**

WorkLifeBalanceID

BalanceLevel

### **Department**

DepartmentID

DepartmentName

**Employee**

Age  
Attrition  
Department  
DistanceFromHome  
Education  
EducationField  
EmployeeNumber  
PerformanceRating  
Gender  
JobLevel  
MaritalStatus  
DailyRate  
HourlyRate  
MonthlyRate  
PercentSalaryHike  
RelationshipSatisfaction  
StockOptionLevel  
OverTime  
StandardHours  
TotalWorkingYears  
TrainingTimeLastYear  
YearsAtCompany  
YearsInCurrentRole  
YearsSinceLastPromotion  
YearsWithCurrManager  
BusinessTravelID  
NumCompaniesWorked  
JobInvolvementID

**DepartmentID**

DistanceFromHome  
EducationID  
EducationField  
EmployeeNumber  
GenderID  
JobLevelID  
JobRoleID  
MaritalStatusID  
OverTime  
StandardHours

## Data Objects

### SQL Queries

#### SQL Query 1

```
CREATE TABLE Gender ( GenderID INT PRIMARY KEY,  
GenderType VARCHAR(50));
```

#### SQL Query 2

```
CREATE TABLE Education ( EducationID INT PRIMARY KEY, EducationLevel VARCHAR(50));
```

#### SQL Query 3

```
CREATE TABLE Department ( DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(50));
```

#### SQL Query 4

```
CREATE TABLE BusinessTravel ( BusinessTravelID INT PRIMARY KEY, TravelType VARCHAR(50));
```

#### SQL Query 5

```
Select * from worklifebalance;
```

#### SQL Query 6

```
select * from maritalstatus;
```

#### SQL Query 7

```
select * from group8.jobinvolvement;
```

#### SQL Query 8

```
select * from group8.relationshipsatisfaction;
```

#### SQL Query 9

```
select * from group8.performancerating;
```

#### SQL Query 10

```
select * from group8.environmentsatisfaction;
```

#### SQL Query 11

```
select * from group8.relationshipsatisfaction;
```

### Query

```
Create Database Group8; use Group8;
```

```
-- Create Education Table
```

```
CREATE TABLE Education ( EducationID INT PRIMARY KEY, EducationLevel VARCHAR(50));
```

```
-- Create EnvironmentSatisfaction Table
```

```
CREATE TABLE EnvironmentSatisfaction ( EnvironmentSatisfactionID INT PRIMARY KEY,  
SatisfactionLevel VARCHAR(50));
```

```
-- Create JobInvolvement Table
```

```
CREATE TABLE JobInvolvement ( JobInvolvementID INT PRIMARY KEY, InvolvementLevel  
VARCHAR(50));
```

```
-- Create JobRole Table
```

```
CREATE TABLE JobRole ( JobRoleID INT PRIMARY KEY, Role VARCHAR(50));
```

```
-- Create JobSatisfaction Table
```

```
CREATE TABLE JobSatisfaction ( JobSatisfactionID INT PRIMARY KEY, SatisfactionLevel VARCHAR(50));
```

```
-- Create MaritalStatus Table
```

```
CREATE TABLE MaritalStatus ( MaritalStatusID INT PRIMARY KEY, Status VARCHAR(50));
```

```
-- Create BusinessTravel Table
```



```

CREATE TABLE BusinessTravel (
BusinessTravelID INT PRIMARY KEY, TravelType VARCHAR(50));

-- Create Gender Table
CREATE TABLE Gender (
GenderID INT PRIMARY KEY,
GenderType VARCHAR(50));

-- Create PerformanceRating Table
CREATE TABLE PerformanceRating ( PerformanceRatingID INT PRIMARY KEY, RatingLevel
VARCHAR(50));

-- Create RelationshipSatisfaction Table
CREATE TABLE RelationshipSatisfaction (
RelationshipSatisfactionID INT PRIMARY KEY, SatisfactionLevel VARCHAR(50));

-- Create WorkLifeBalance Table
CREATE TABLE WorkLifeBalance ( WorkLifeBalanceID INT PRIMARY KEY, BalanceLevel VARCHAR(50));

-- Create Department Table
CREATE TABLE Department ( DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(50));

-- EXPERIENCE
CREATE TABLE Experience( EmployeeNumber INT PRIMARY KEY,
TotalWorkingYears INT, TrainingTimesLastYear INT,
YearsAtCompany INT, YearsInCurrentRole INT, YearsSinceLastPromotion INT, YearsWithCurrManager
INT, BusinessTravelID INT, NumCompaniesWorked INT,
FOREIGN KEY (BusinessTravelID) REFERENCES BusinessTravel(BusinessTravelID));

-- job review
CREATE TABLE JobReviewDetails( EmployeeNumber INT PRIMARY KEY,JobInvolvementID INT,
JobSatisfactionID INT , WorkLifeBalanceID INT, EnvironmentSatisfactionID INT,
FOREIGN KEY (EnvironmentSatisfactionID) REFERENCES
EnvironmentSatisfaction(EnvironmentSatisfactionID),
FOREIGN KEY (WorkLifeBalanceID) REFERENCES WorkLifeBalance(WorkLifeBalanceID),
FOREIGN KEY (JobInvolvementID) REFERENCES JobInvolvement(JobInvolvementID));

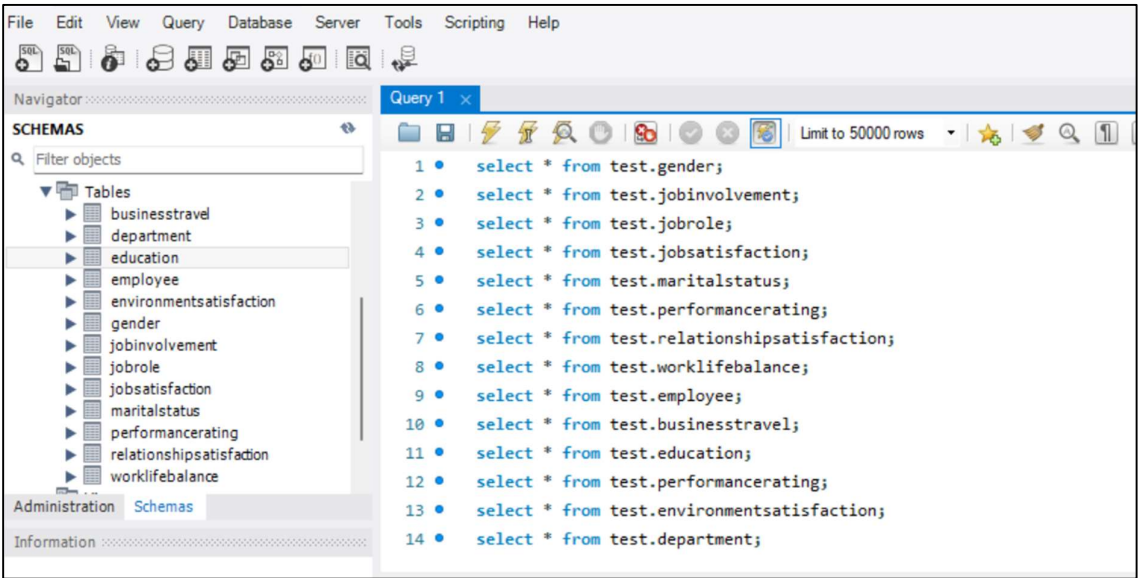
CREATE TABLE SalaryDetails ( EmployeeNumber INT PRIMARY KEY, DailyRate INT,
HourlyRate INT, MonthlyIncome INT, MonthlyRate
INTYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY,
PercentSalaryHike INT, StockOptionLevel INT);

-- Create Employee Table

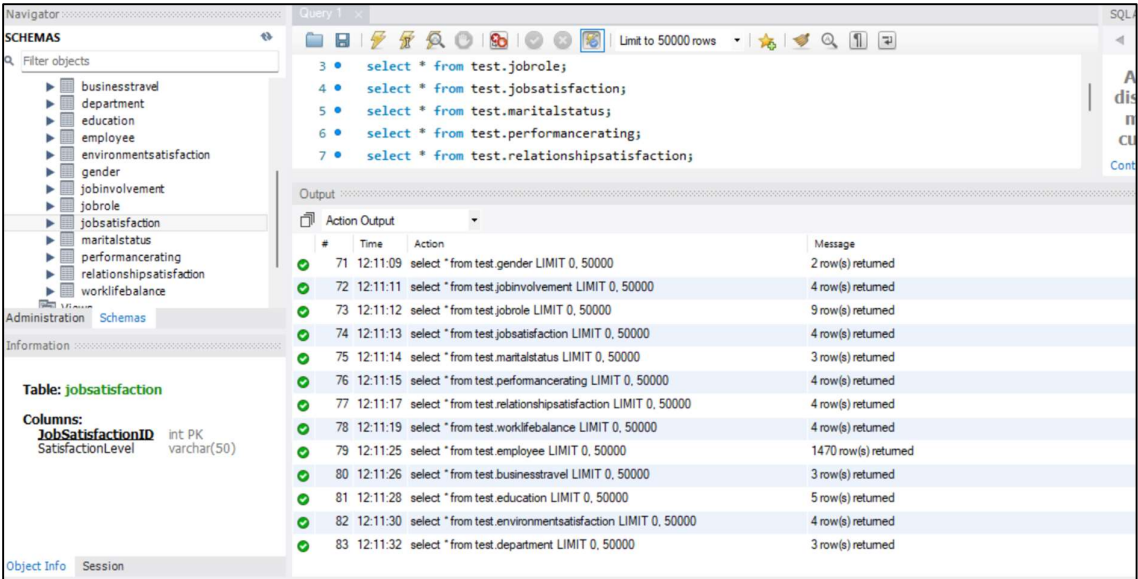
CREATE TABLE Employee (Age INT,
Attrition VARCHAR(10), DepartmentID INT, DistanceFromHome INT, EducationID INT, EducationField
VARCHAR(50),EmployeeNumber INT PRIMARY KEY, GenderID INT,
JobLevelID INT, JobRoleID INT, MaritalStatusID INT, OverTime VARCHAR(50), StandardHours INT,
FOREIGN KEY (EducationID) REFERENCES Education(EducationID), FOREIGN KEY (JobRoleID)
REFERENCES JobRole(JobRoleID),
FOREIGN KEY (MaritalStatusID) REFERENCES MaritalStatus(MaritalStatusID));

```

SQL:



Count of tables:



**Details:**

- gender table consists of 2 rows.
- Jobinvolvement table consist of 4 rows
- jobrole table consist of 9 rows
- jobsatisfaction table consist of 4 rows
- maritalstatus table consist of 3 rows
- performancerating table consist of 4 rows
- relationshipsatisfaction table consist of 4 rows
- worklifebalance table consist of 4 rows
- employee table consist of 1470 rows
- businessstravel table consist of 3 rows
- education table consist of 5 rows
- environmentsatisfaction table consist of 4 rows
- department table consist of 3 rows

## SQL Queries

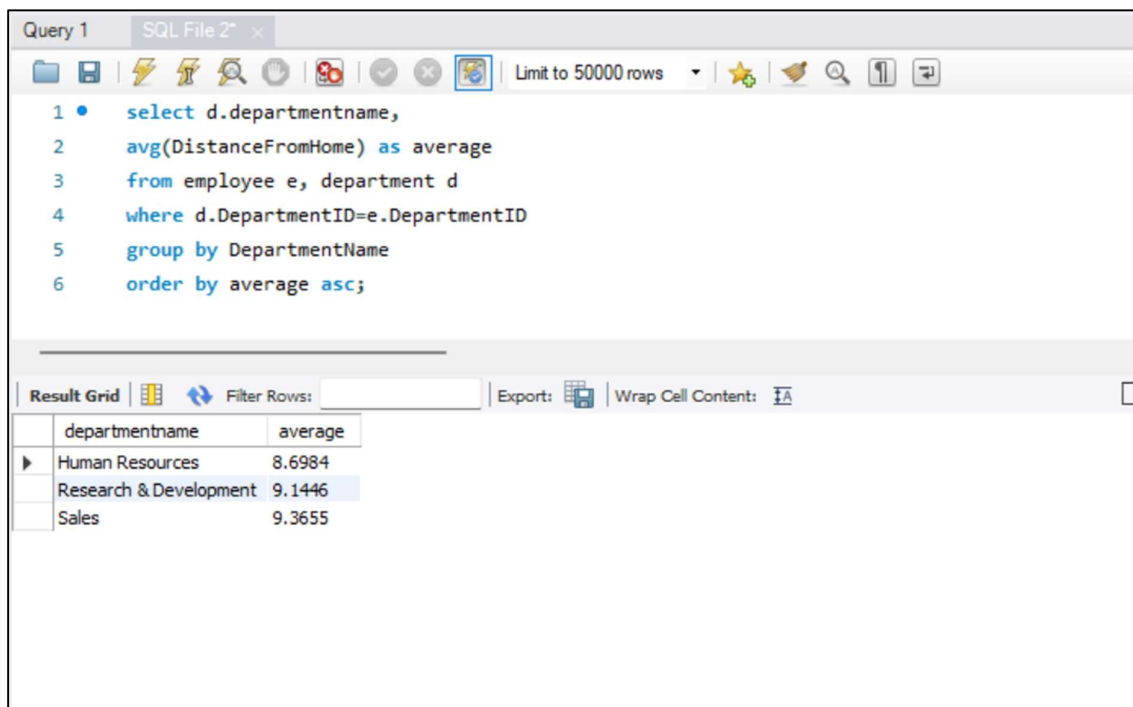
### Question 2

Which department's employee is the most likely to have the shortest commute between home and work?

**Translation:** Select department name and the average distance from home for employees within each department from the employee and department tables using the DepartmentID column as a reference. Group the results by DepartmentName and calculate the average distance from home for each group. Finally, order the results in ascending order based on the calculated average.

### SQL Query

```
select d.departmentname,  
avg(DistanceFromHome) as average  
from employee e, department d  
where d.DepartmentID=e.DepartmentID  
group by DepartmentName  
order by average asc;
```



The screenshot shows a SQL query editor window titled "Query 1" with a toolbar at the top. The query text is as follows:

```
1 • select d.departmentname,  
2   avg(DistanceFromHome) as average  
3   from employee e, department d  
4   where d.DepartmentID=e.DepartmentID  
5   group by DepartmentName  
6   order by average asc;
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the query. The results are shown in a table with two columns: "departmentname" and "average". The data is sorted in ascending order of the average distance.

departmentname	average
Human Resources	8.6984
Research & Development	9.1446
Sales	9.3655

**Interpretation:** Upon computation of the average commuting distance for each department, it was determined that the Human Resource Department boasts the shortest commute between home and work. Therefore, Human resources department's employee is the most likely to have the shortest commute between home and work.

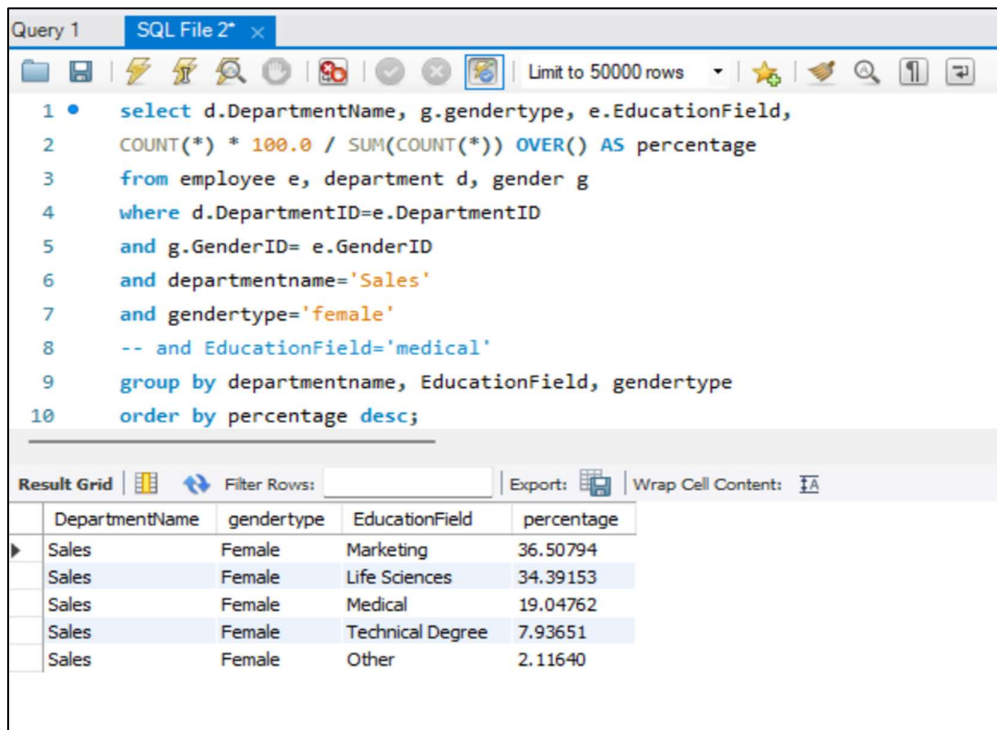
### Question 3

A new employee from a Medical-related education field wants to work in Sales. Do you believe the company might be able to give her a chance to work in Sales? Why or Why not?

**Translation:** Select department name, gender type, and education field, along with the percentage of employees belonging to the 'Sales' department who are 'female' from the employee, department, and gender tables using DepartmentID and GenderID. Filter the results to include only the 'Sales' department and 'female' gender type. Group the results by department name, education field, and gender type. Calculate the percentage within each group relative to the total count of employees in the 'Sales' department who are female. Finally, order the results in descending order based on the calculated percentage.

### SQL Query

```
select d.DepartmentName, g.gendertype, e.EducationField,
COUNT(*) * 100.0 / SUM(COUNT(*)) OVER() AS percentage
from employee e, department d, gender g
where d.DepartmentID=e.DepartmentID
and g.GenderID= e.GenderID
and departmentname='Sales'
and gendertype='female'
-- and EducationField='medical'
group by departmentname, EducationField, gendertype
order by percentage desc;
```



The screenshot shows a SQL query window titled 'Query 1' with a toolbar and a 'Limit to 50000 rows' dropdown. The query text is as follows:

```
1 • select d.DepartmentName, g.gendertype, e.EducationField,
2   COUNT(*) * 100.0 / SUM(COUNT(*)) OVER() AS percentage
3   from employee e, department d, gender g
4   where d.DepartmentID=e.DepartmentID
5   and g.GenderID= e.GenderID
6   and departmentname='Sales'
7   and gendertype='female'
8   -- and EducationField='medical'
9   group by departmentname, EducationField, gendertype
10  order by percentage desc;
```

Below the query editor is the 'Result Grid' tab, which displays the following data:

DepartmentName	gendertype	EducationField	percentage
Sales	Female	Marketing	36.50794
Sales	Female	Life Sciences	34.39153
Sales	Female	Medical	19.04762
Sales	Female	Technical Degree	7.93651
Sales	Female	Other	2.11640

**Interpretation:** The query results show that the Sales department mostly recruits females from Marketing (36.5%) and Life Sciences (34.4%) backgrounds, while those with medical education make up only 19%, indicating lower recruitment from this field. This suggests Marketing professionals may have skills useful for Sales, possibly explaining their higher representation. Thus, her chances of being offered a sales department role seem unlikely due to her medical education background.

#### Question 4

The HR department feels they have the highest job satisfaction while Research & Development department feels their department has the highest environment satisfaction. Who is right?

**Translation:** Select department name, the average job satisfaction, and the average environment satisfaction for employees within each department from employee, department, job satisfaction, and environment satisfaction tables using the departmentID and jobsatisfactionID, and EnvironmentSatisfactionID. Group the results by department name and calculate the average job satisfaction and environment satisfaction for each group. Finally, order the results in descending order based on the calculated average job satisfaction.

#### SQL Query

```
select d.DepartmentName,  
Avg(j.jobSatisfactionID) as jobsatisfaction,  
Avg(es.environmentSatisfactionID) as environsatis  
from employee e, department d, jobsatisfaction j, environmentsatisfaction es  
where d.DepartmentID=e.DepartmentID  
and j.JobSatisfactionID= e.JobSatisfactionID  
and es.EnvironmentSatisfactionID=e.EnvironmentSatisfactionID  
-- and departmentname in ('Human Resources','Research & Development')  
group by departmentname  
order by jobsatisfaction desc;
```

The screenshot shows a SQL IDE interface. On the left, a 'SCHEMAS' pane lists database objects, with 'environmentsatisfaction' selected. The main area displays a SQL query (Query 1) and its results in a 'Result Grid'. The query calculates the average job satisfaction and environment satisfaction for each department, filtering for 'Human Resources' and 'Research & Development'. The results show three departments: Sales, Research & Development, and Human Resources, with their respective average job satisfaction and environment satisfaction values.

DepartmentName	jobsatisfaction	environsatis
Sales	2.7511	2.6794
Research & Development	2.7263	2.7440
Human Resources	2.6032	2.6825

**Interpretation:** The Research & Development department is correct in asserting that they have the highest environment satisfaction. Conversely, the HR department's assertion of having the highest job satisfaction is incorrect, as the Sales department actually has the highest job satisfaction.

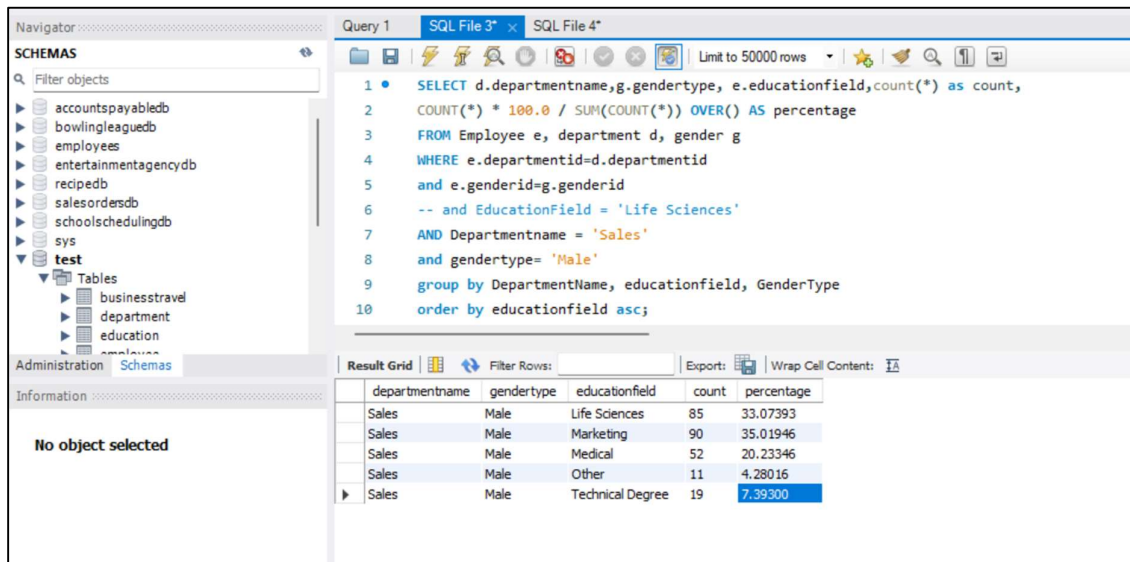
### Question 9

A new employee from a Life Sciences education field wants to work in Sales. Do you believe the company might be able to give him a chance to work in Sales? Why or Why not?

**Translation:** Select department name, gender type, education field, count of employees, and the percentage of employees within the 'Sales' department who are 'Male'. Associate employees with their respective departments using the DepartmentID and genderID column as a reference, and also join the gender table using the GenderID column. Filter the results to include only employees in the 'Sales' department who are 'Male'. Group the results by DepartmentName, education field, and GenderType. Calculate the count and percentage within each group, and finally, order the results in ascending order based on the education field.

### SQL Query

```
COUNT(*) * 100.0 / SUM(COUNT(*)) OVER() AS percentage
FROM Employee e, department d, gender g
WHERE e.departmentid=d.departmentid
and e.genderid=g.genderid
-- and EducationField = 'Life Sciences'
AND Departmentname = 'Sales'
and gendertype= 'Male'
group by DepartmentName, educationfield, GenderType
order by educationfield asc;
```



The screenshot shows a SQL query editor with a query window and a results grid. The query is as follows:

```
1 • SELECT d.departmentname,g.gendertype, e.educationfield,count(*) as count,
2 COUNT(*) * 100.0 / SUM(COUNT(*)) OVER() AS percentage
3 FROM Employee e, department d, gender g
4 WHERE e.departmentid=d.departmentid
5 and e.genderid=g.genderid
6 -- and EducationField = 'Life Sciences'
7 AND Departmentname = 'Sales'
8 and gendertype= 'Male'
9 group by DepartmentName, educationfield, GenderType
10 order by educationfield asc;
```

The results grid shows the following data:

departmentname	gendertype	educationfield	count	percentage
Sales	Male	Life Sciences	85	33.07393
Sales	Male	Marketing	90	35.01946
Sales	Male	Medical	52	20.23346
Sales	Male	Other	11	4.28016
Sales	Male	Technical Degree	19	7.39300

**Interpretation:** The company's data shows that employees from the Life Sciences education field make up 33% of the Sales department, indicating a reasonable presence. While Marketing has a slightly higher representation at 35%, the company's willingness to hire from diverse educational backgrounds suggests that they might consider giving the new Life Sciences employee a chance to work in Sales. This is further supported by their strong likelihood of being placed in the sales department, providing an additional reason for consideration.

### Question 10

HR feels that their environment satisfaction score is higher than Sales but HR job satisfaction score is lower than Research & Development. Are they right?

**Translation:** Select department name, average job satisfaction, and average environment satisfaction for employees within each department from the employee, department, job satisfaction, and environment satisfaction tables using DepartmentID, JobsatisfactionID, and EnvironmentsatisfactionID. Group the results by DepartmentName and calculate the average job satisfaction and environment satisfaction for each group. Finally, order the results in descending order based on the calculated average job satisfaction.

### SQL Query

```
select d.DepartmentName,  
Avg(j.jobSatisfactionID) as jobsatisfaction,  
Avg(es.environmentSatisfactionID) as environsatis  
from employee e, department d, jobsatisfaction j, environmentsatisfaction es  
where d.DepartmentID=e.DepartmentID  
and j.JobSatisfactionID= e.JobSatisfactionID  
and es.EnvironmentSatisfactionID=e.EnvironmentSatisfactionID  
group by departmentname  
order by jobsatisfaction desc;
```

The screenshot shows a SQL IDE interface. On the left, the 'SCHEMAS' pane lists various tables including 'businesstravel', 'department', 'education', 'employee', 'environmentsatisfaction', 'gender', 'jobinvolvement', 'jobrole', 'jobsatisfaction', 'maritalstatus', 'performancering', 'relationshipsatisfaction', and 'worklifebalance'. The 'employee' table is selected. The main pane displays a SQL query (Query 1) that calculates the average job satisfaction and environment satisfaction for each department. The query is as follows:

```
1 select d.DepartmentName,  
2 Avg(j.jobSatisfactionID) as jobsatisfaction,  
3 Avg(es.environmentSatisfactionID) as environsatis  
4 from employee e, department d, jobsatisfaction j, environmentsatisfaction es  
5 where d.DepartmentID=e.DepartmentID  
6 and j.JobSatisfactionID= e.JobSatisfactionID  
7 and es.EnvironmentSatisfactionID=e.EnvironmentSatisfactionID  
8 group by departmentname  
9 order by jobsatisfaction desc;  
10
```

Below the query, the 'Result Grid' shows the following data:

DepartmentName	jobsatisfaction	environsatis
Sales	2.7511	2.6794
Research & Development	2.7263	2.7440
Human Resources	2.6032	2.6825

The 'Information' pane on the left shows details for the 'environmentsatisfaction' table, including columns 'EnvironmentSatisfactionID' (int PK) and 'SatisfactionLevel' (varchar(50)).

**Interpretation:** The HR department's belief regarding their environment satisfaction score appears to be correct, as their score of 2.6825 is indeed slightly higher than that of the Sales department, which is 2.6794. However, their belief that their job satisfaction score is lower than Research & Development is also accurate, with HR scoring 2.6032 compared to Research & Development's higher score of 2.7263. Therefore, HR's perception aligns with the provided scores for both environment and job satisfaction.



Mongodb:

cluster0.bkwyw...

Documents  
test.employee

My Queries

Databases

Search

admin

local

test

employee

test.employee

1.5k  
DOCUMENTS

1  
INDEXES

Documents

Aggregations

Schema

Indexes

Validation

Filter

Type a query: { field: 'value' } or [Generate query](#)

Explain

Reset

Find

Options

ADD DATA

EXPORT DATA

1 - 20 of 1470

```
{
  "_id": ObjectId('6571697aab500673c0820eb7'),
  "Age": 41,
  "Attrition": "Yes",
  "BusinessTravel": "Travel_Rarely",
  "DailyRate": 1162,
  "Department": "Sales",
  "DistanceFromHome": 1,
  "Education": 2,
  "EducationField": "Life Sciences",
  "EmployeeCount": 1,
  "EmployeeNumber": 1,
  "EnvironmentSatisfaction": 2,
  "Gender": "Female",
  "HourlyRate": 94,
  "JobInvolvement": 3
}
```

## Mongo Queries

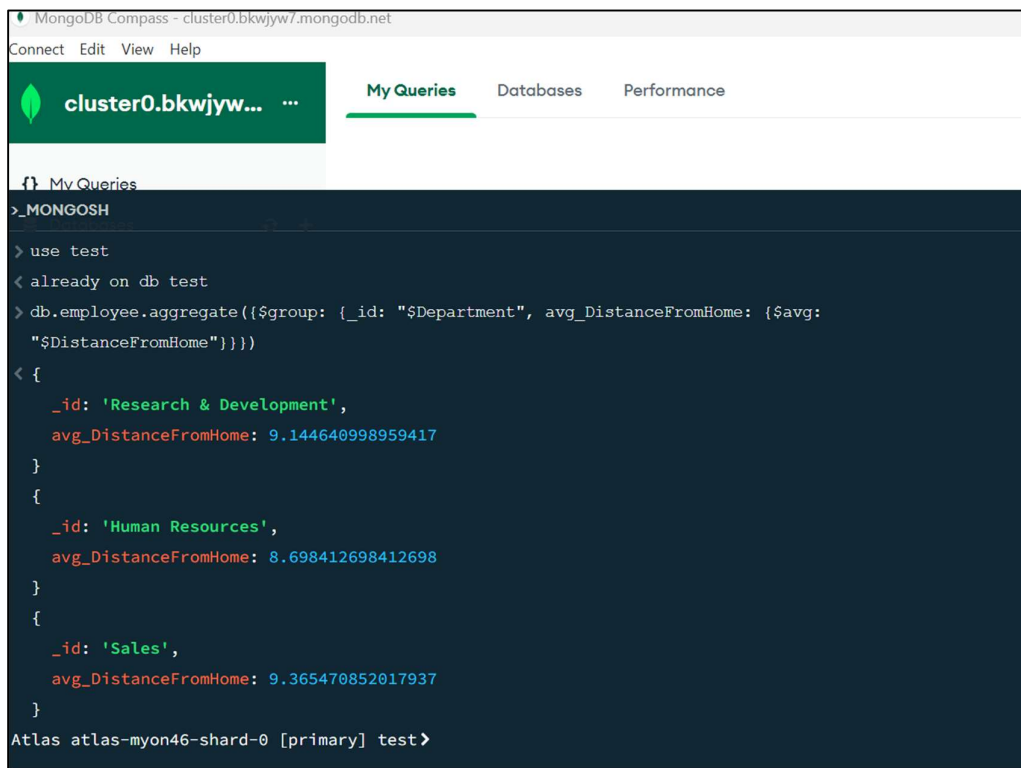
### Question 8

The company has been paying gas expenses for miles traveled by employees between their home and work. If they want to increase the per mile compensation, which department's employees will gain the most?

**Translation:** Retrieve the department name and the average distance from home for employees within each department from the "employee" collection. Group the results by the "Department" field and calculate the average distance from home using the \$avg aggregation operator.

### Query

```
db.employee.aggregate({$group: {_id: "$Department", avg_DistanceFromHome: {$avg: "$DistanceFromHome"}}})
```



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'cluster0.bkwjyw.mongodb.net'. The 'My Queries' tab is active. The query editor shows the following aggregation query:

```
{ $group: { _id: "$Department", avg_DistanceFromHome: { $avg: "$DistanceFromHome" } } }
```

The results are displayed in a JSON array format:

```
[ { _id: 'Research & Development', avg_DistanceFromHome: 9.144640998959417 }, { _id: 'Human Resources', avg_DistanceFromHome: 8.698412698412698 }, { _id: 'Sales', avg_DistanceFromHome: 9.365470852017937 } ]
```

The status bar at the bottom shows 'Atlas atlas-myon46-shard-0 [primary] test'.

**Interpretation:** To determine which department's employees will benefit the most from an increase in per-mile compensation for gas expenses, we need to consider the department where employees have the longest average commuting distance from home to work. In this case, it's the Sales department, where employees have an average commuting distance of 9.36 miles. Therefore, increasing the per-mile compensation would likely benefit the Sales department's employees the most, as they travel the farthest on average.

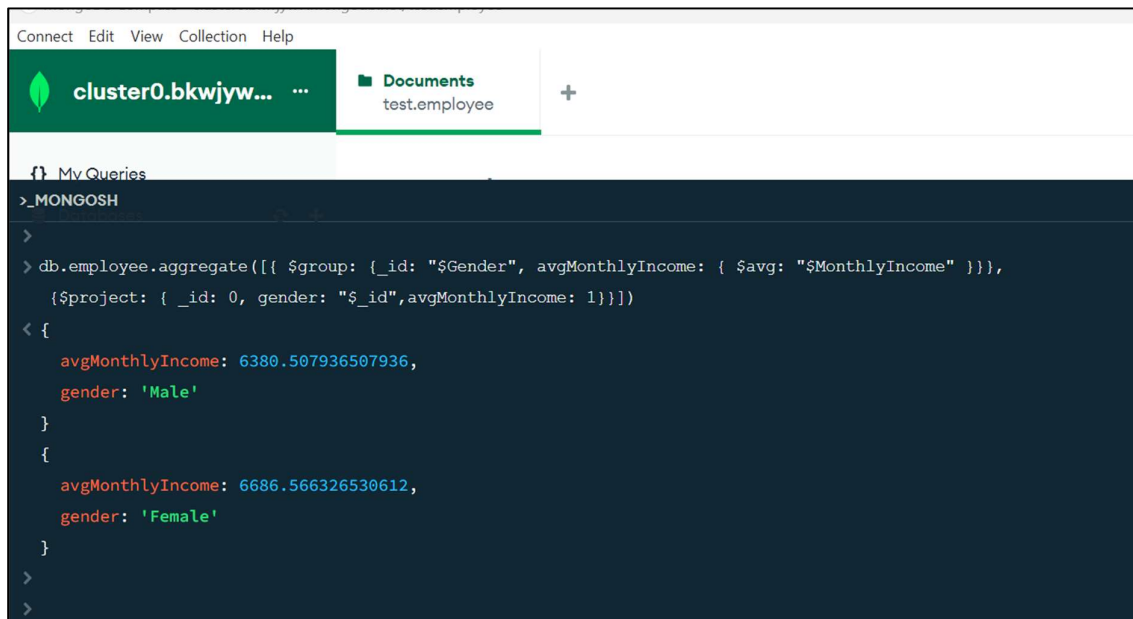
### Question 5

An employee in Sales department has complained to HR saying that females are paid less than males in the company, in all departments. What insight can you provide to prove or disprove that statement?

**Translation:** Retrieve the gender and the average monthly income for employees within each gender category from the "employee" collection. Group the results by the "Gender" field and calculate the average monthly income using the \$avg aggregation operator. Then, use \$project to reshape the output to include only the "gender" and "avgMonthlyIncome" fields while excluding the default "\_id" field.

### Query

```
db.employee.aggregate([{$group: {_id: "$Gender", avgMonthlyIncome: { $avg: "$MonthlyIncome" } }},
  {$project: { _id: 0, gender: "$_id", avgMonthlyIncome: 1}}])
```



The screenshot shows a MongoDB query interface. The top bar includes 'Connect', 'Edit', 'View', 'Collection', and 'Help' menus. Below this, there's a green sidebar with a leaf icon and the text 'cluster0.bkwjyw...'. To the right of the sidebar, there's a 'Documents' tab with a plus sign and the text 'test.employee'. The main area is a dark blue console with a light blue header 'My Queries'. Below the header, the prompt '>\_MONGOSH' is followed by the query: 

```
> db.employee.aggregate([{$group: {_id: "$Gender", avgMonthlyIncome: { $avg: "$MonthlyIncome" } }},
  {$project: { _id: 0, gender: "$_id", avgMonthlyIncome: 1}}])
```

 The results are displayed as a JSON array: 

```
< [
  {
    avgMonthlyIncome: 6380.507936507936,
    gender: 'Male'
  },
  {
    avgMonthlyIncome: 6686.566326530612,
    gender: 'Female'
  }
]
```

**Interpretation:** Based on the provided data, it appears that, on average, females are actually paid more than males in the company across all departments. The average monthly income for males is \$6380.5, whereas for females, it is higher at \$6686.57. This information disproves the employee's complaint, indicating that females receive higher compensation on average across the company's departments.

### Question 11

An employee from Medical education field working in Sales department has spread a rumor saying that employees with his educational background are paid more in Research & Development than in Sales. What insight can you provide to prove or disprove that statement?

**Translation:** Retrieve the department name, education field (specifically "Medical"), and the average salary for employees with "Medical" education in the "Sales" and "Research & Development" departments from the "Employee" collection. First, filter the employees with "Medical" education in the specified departments. Then, group the results by the "Department" and "EducationField" fields and calculate the average salary using the \$avg aggregation operator.

### Query

```
db.Employee.aggregate([ { $match: { EducationField: "Medical", Department: { $in: ["Sales", "Research & Development"] } } }, { $group: { _id: { Department: "$Department", EducationField: "$EducationField" }, avg_Salary: { $avg: "$MonthlyIncome" } } } ] );
```

```
>_MONGOSH
}
> db.employee.aggregate([
  { $match: { EducationField: "Medical", Department: { $in: ["Sales", "Research & Development"] } } },
  { $group: { _id: { Department: "$Department", EducationField: "$EducationField" },
    avg_Salary: { $avg: "$MonthlyIncome" } } } ] );
< {
  _id: {
    Department: 'Research & Development',
    EducationField: 'Medical'
  },
  avg_Salary: 6539.2231404958675
}
{
  _id: {
    Department: 'Sales',
    EducationField: 'Medical'
  },
  avg_Salary: 6377.227272727273
}
Atlas atlas-myon46-shard-0 [primary] test>
```

**Interpretation:** Based on the provided data, it appears that employees with a Medical education background working in Research & Development have a slightly higher average salary of \$6539.22 compared to those in the Sales department with an average salary of \$6377.22. This suggests that the rumor is true, as individuals with a Medical education background in Research & Development do indeed earn slightly more on average than their counterparts in the Sales department.