

1) Our team consists of me (Mehmet Kadri Gofrahlar) and Hasan Ali Özkan.

2)

a) I liked the clarity of this dataset and it gives me an idea about a company structure. Also I thought preparing an ER Diagram for this dataset would be fun and informative.

b) Nine files exist in our data.

c) Our data contains 3311 rows and 31 columns.

d) There are sixteen string (non-numeric) columns in our data.

e) The link of our project's raw data is:

<https://www.kaggle.com/davidepolizzi/hr-data-set-based-on-human-resources-data-set>

f) Bitbucket URL of our ER diagram is:

<https://bitbucket.org/U180709005/db2021/src/master/Project/>

3) Here's our list of the English questions that we would like our system to be able to answer:

a) "Show the names of employees whose performance scores is "needs improvements" and satisfaction scores are lower than the average"

b) "Show the names and positions of employees that started working in this company in 2015"

c) "Show the managers grouped by their crew members, sorted by their crew member count"

d) "Show the names of employees that haven't started to work and needs improvement"

e) "Show the name, pay rate and department of employees whose engagement survey result is low and terminated for more money"

f) "Show the names, pay rates and positions whose pay rate is lower than the average"

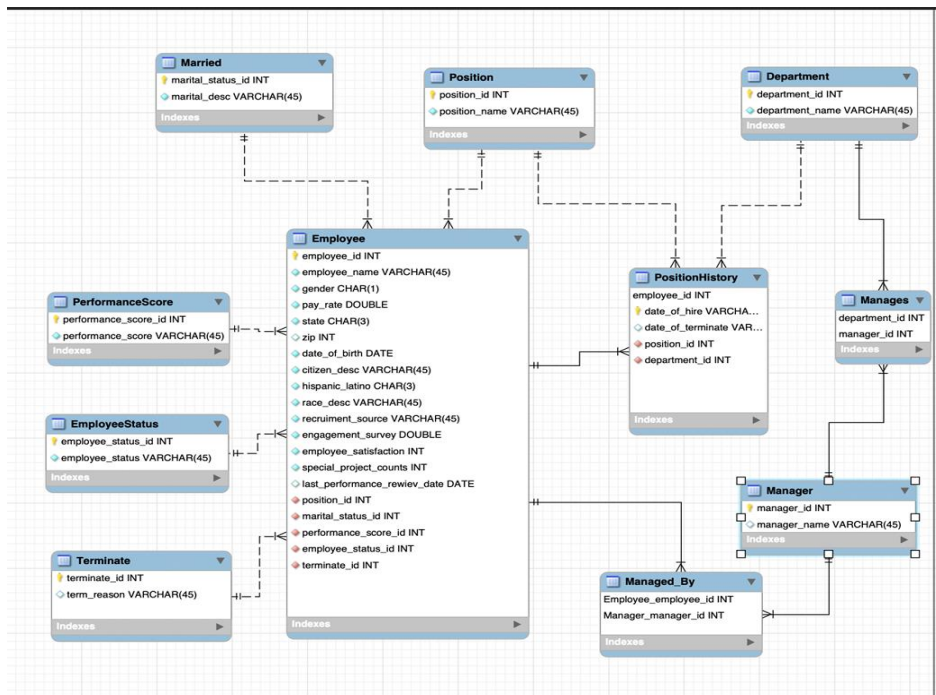
g) "Show the count of employees grouped by department"

h) "Show the names, engagement survey scores and performance scores of the employees whose recruitment source are employee referral"

i) "Show the names and positions of the employees who are older than 60"

j) "Show the names, marital statuses and the departments of the employees that are active"

4)



5) Here are the SQL statements that will implement the English questions from our target queries:

a)

```
"select employee_name
from Employee
where employee_satisfaction < (
    select avg(employee_satisfaction) from Employee)
and employee_id in(
    select performance_score_id
    from PerformanceScore
    where performance_score = "Needs Improvements");"
```

b)

```
"select employee_name, position_name
from Employee as e join Position as p on (e.position_id = p.position_id)
where e.employee_id in (
    select employee_id from Employee as E join PositionHistory P
    on(E.employee_id == P.employee_id)
where date_of_hire between 01.01.2015 and 31.12.2015);"
```

c)

```
"select T.employee_name, m.manager_name, T.total from Manager as m join(
    select Employee_name, E.employee_id ,M.manager_id, count(*) as total
    from Employee as E join ManagedBy as M on(
        M.employee_id = E.employee_id))as T on(
        m.manager_id = T.manager_id)
group by manager_name;"
```

d)

```
"select employee_name
from Employee
where performance_score_id in(
    select performance_score_id
    from PerformanceScore
    where performance_score = "Needs Improvements")
and
employee_status_id in(
    select employee_status_id
    from EmployeeStatus
    where employee_status = "Future Start");"
```

e)

```
"select employee_name, pay_rate
from Employee
where engagement_survey<(select avg(engagement_survey)from Employee)
and terminate_id in(
    select terminate_id
    from Terminate
    where term_reason = "more money");"
```

f)

```
"select employee_name, position_name
from Employee as e left join Position as p on(e.position_id = p.position_id)
where e.pay_rate<(select avg(pay_rate) from Employee);"
```

g)

```
select d.department_name, count(T.employee_name) as employee_count
from Department as d
join(select Employee.employee_name, PositionHistory.department_id from
      Employee join PositionHistory on
      (Employee.employee_id = PositionHistory.employee_id)) as T
on(T.department_id = d.department_id)
group by employee_count;
```

h)

```
select e.employee_name ,e.engagement_survey ,p.performance_score
from Employee as e join PerformanceScore as p
on(e.performance_score_id = p.performance_score_id)
where recruitment_source = "Employee Referral";
```

i)

```
"select employee_name, position_name
from Employee as e left join Position as p on(e.position_id= p.position_id)
where e.date_of_birth<1960;"
```

j)

```
"select employee_name, department_name, marital_status
from Department
where department_id, employee_name, marital_status in(
select department_id, employee_name, marital_status
from Employee as e join PositionHistory p on (e.employee_id = p.employee_id)
where employee_name, employee_id ,marital_status in(
select employee_name, employee_id, marital_status
from Employee
where employee_name, marital_status in (
select employee_name, marital_status from Employee as e join
MaritalStatus as m on(m.marital_status_id = e.marital_status_id))
and employee_status_id in(
select employee_status_id
from EmployeeStatus
where employee_status = "Active"))));"
```

6) We've downloaded the data from "<https://www.kaggle.com/davidepolizzi/hr-data-set-based-on-human-resources-data-set>" as a .txt file. Then we divided it into tables using excel and converted it to a ".xlsx" file. As the final step, we will be inserting the data using MySQL Workbench.

7) We won't be implementing any front-end user interface because of 2 reasons. First reason is, we don't know how to do it yet. And second reason is that we didn't find implementing any front-end user interface reasonable for our data because we think it wouldn't be efficient. We probably would use this database only to show it to the boss or the H.R. department. Therefore implementing a user interface just for 2 people didn't make sense. We will be writing one view and one stored-procedure (IN , OUT , IN-OUT).

8) MySQL 8.0.22 on my personal computer Intel i7 with 4 GB of memory.