DATABASE MANAGEMENT PROJECT PROPOSAL

The key to success in any organization is attracting and retaining top talent. However, the ongoing trend of employee attrition i.e the gradual reduction in employee numbers. Employee attrition happens when the size of your workforce diminishes over time. This means that employees are leaving faster than they are hired.

The employee Retention System is designed to make it easier for the company to enhance employee retention rate. This will be done by designing a system with both Employee data and HR data that will bring up aore insight in the way that employee information can be used. We will be creating the database using MySQL for this.

Project Aim :

The Aim of this project is to develop a system that enhances a company's employee retention rate.

Projective Objective :

The Objective of this project is to create a system that helps to understand the reasoning behind employee attrition and plan better retention plans.

SL. No	Topics	Deadline	Contributor
1	Database Design	10/07/2023	Ajith kumar Sukumar, Yashwanth reddy Solipuram
2	ER Diagram	10/07/2023	Ajith kumar Sukumar, Yashwanth reddy Solipuram
3	Table Creation • Employee Details • Employee Profile • Department Details • Job Details • Employee Satisfaction Rating • Employee Absenteeism Factors • Payroll Data	10/14/2023	Ajith kumar Sukumar Yashwanth reddy Solipuram Ajith kumar Sukumar Yashwanth reddy Solipuram Ajith kumar Sukumar Yashwanth reddy solipuram Ajith kumar Sukumar
4	Data Generation	10/21/2023	Ajith kumar Sukumar, Yashwanth reddy Solipuram
5	Normalization	10/28/2023	Ajith kumar Sukumar, Yashwanth reddy Solipuram
6	Finding Insights (Query)	11/04/2023	Ajith kumar Sukumar, Yashwanth reddy Solipuram
7	Documentation	11/11/2023	Ajith kumar Sukumar, Yashwanth reddy Solipuram

SUMMARY

Purpose:

This document's goal is to provide a thorough explanation of the design procedure used to create and construct a database for the Employee Retention System. The properties and entities included in the database are all described on this page. The ultimate objective is to create a database for a Company that allows for seamless access to any type of data that is needed for a specific Employee profile, job satisfaction, Payroll data, Employee satisfaction rating, or business use case, regardless of how complex the use case may be. This will ensure the quickest data retrieval time and instant insights from the available data by utilizing preloaded intelligent analytics and data visualization.

Business Rules:

- One employee should have only one Job ID, Job title.
- One Employee can be in different Departments.
- One Employee should be assigned under one manager only.
- Manager Can have multiple Employees assigned under one manager.

ENTITY RELATIONSHIP DIAGRAM REPRESENTING DATABASE DESIGN :

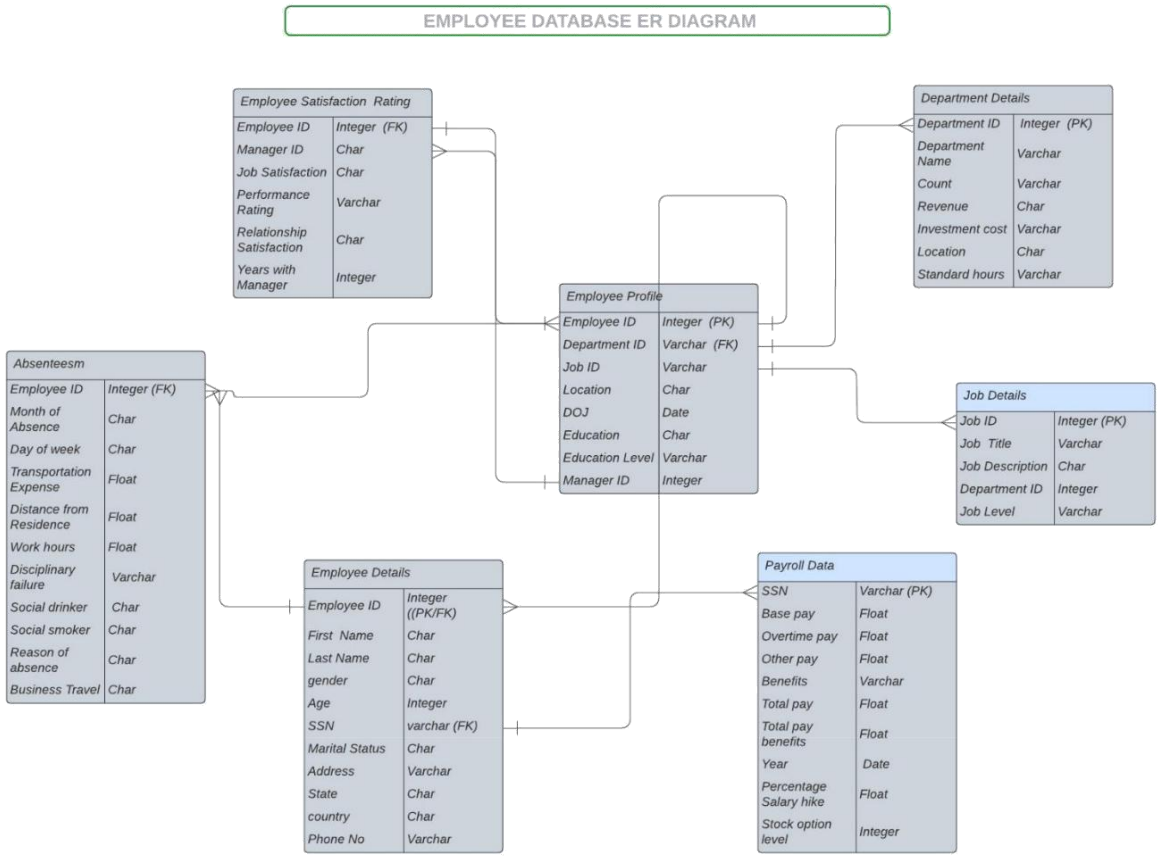
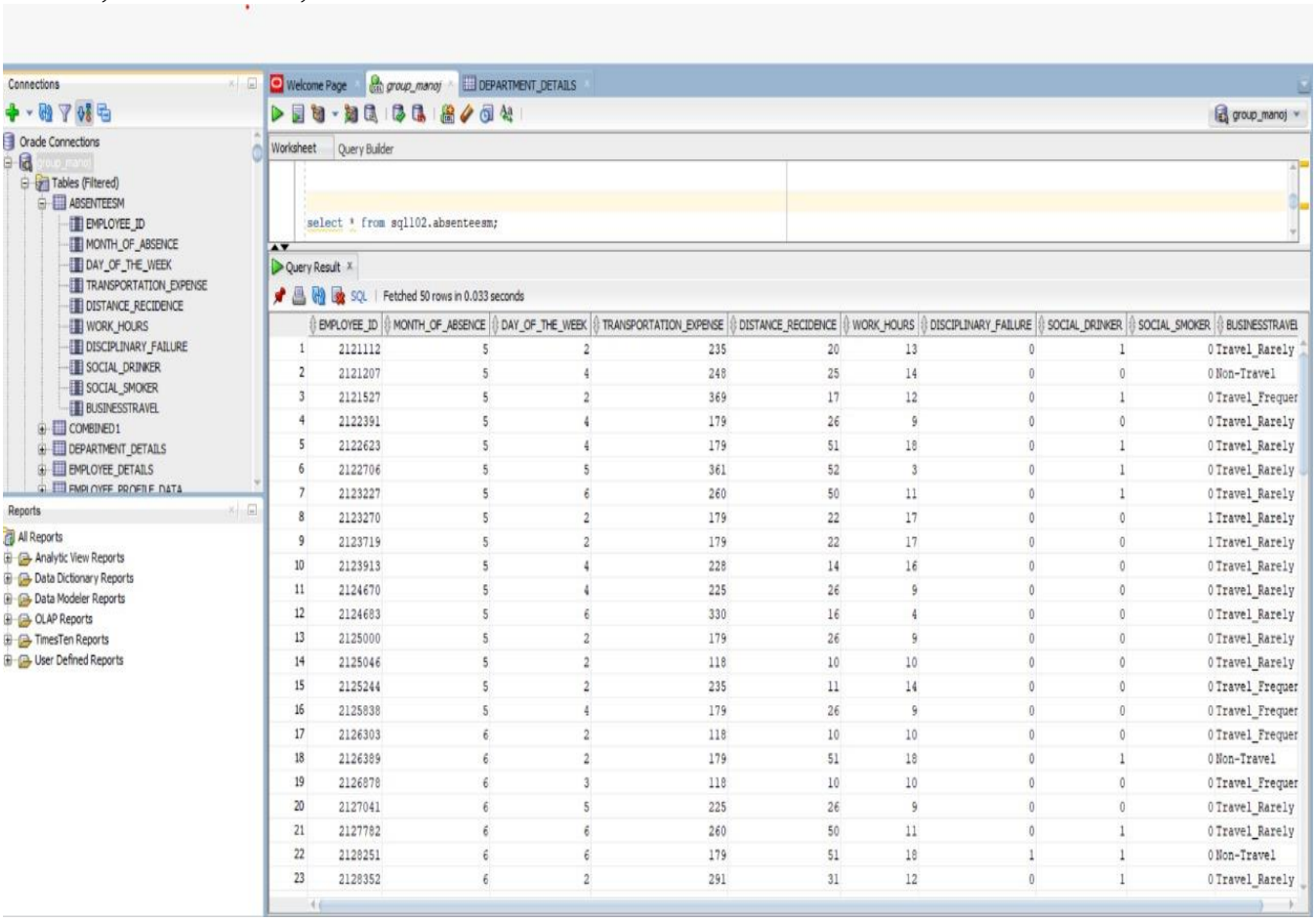

EMPLOYEE DATABASE ER DIAGRAM
Employee Satisfaction Rating
Employee ID Integer (FK)
Manager ID Char
Job Satisfaction Char
Performance Rating Varchar
Relationship Satisfaction Char
Years with Manager Integer
Department Details
Department ID Integer (PK)
Department Name Varchar
Count Varchar
Revenue Char
Investment cost Varchar
Location Char
Standard hours Varchar
Employee Profile
Employee ID Integer (PK)
Department ID Varchar (FK)
Job ID Varchar
Location Char
DOJ Date
Education Char
Education Level Varchar
Manager ID Integer
Absenteesm
Employee ID Integer (FK)
Month of Absence Char
Day of week Char
Transportation Expense Float
Distance from Residence Float
Work hours Float
Disciplinary failure Varchar
Social drinker Char
Social smoker Char
Reason of absence Char
Business Travel Char
Job Details
Job ID Integer (PK)
Job Title Varchar
Job Description Char
Department ID Integer
Job Level Varchar
Employee Details
Employee ID Integer ((PK/FK)
First Name Char
Last Name Char
gender Char
Age Integer
SSN varchar (FK)
Marital Status Char
Address Varchar
State Char
country Char
Phone No Varchar
Payroll Data
SSN Varchar (PK)
Base pay Float
Overtime pay Float
Other pay Float
Benefits Varchar
Total pay Float
Total pay benefits Float
Year Date
Percentage Salary hike Float
Stock option level Integer

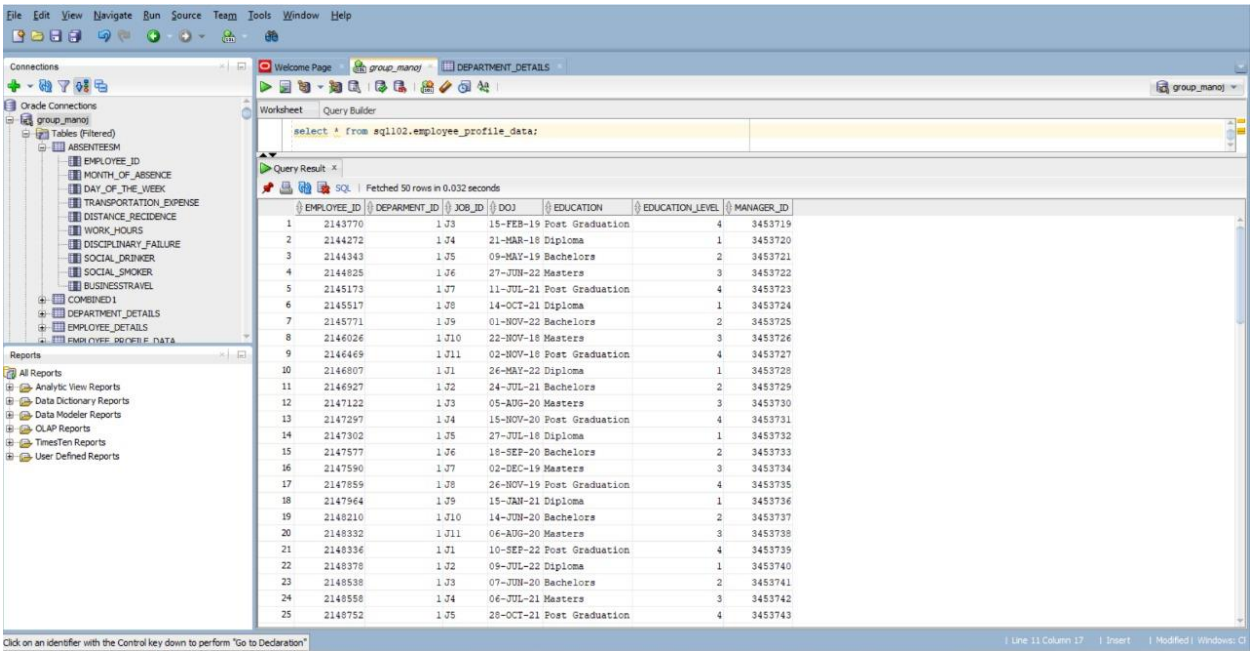
Table Views:

Absenteesm: This table has information about an Employee ID, Month of absence, Reasons for absence, Social Drinker, Social Smoker



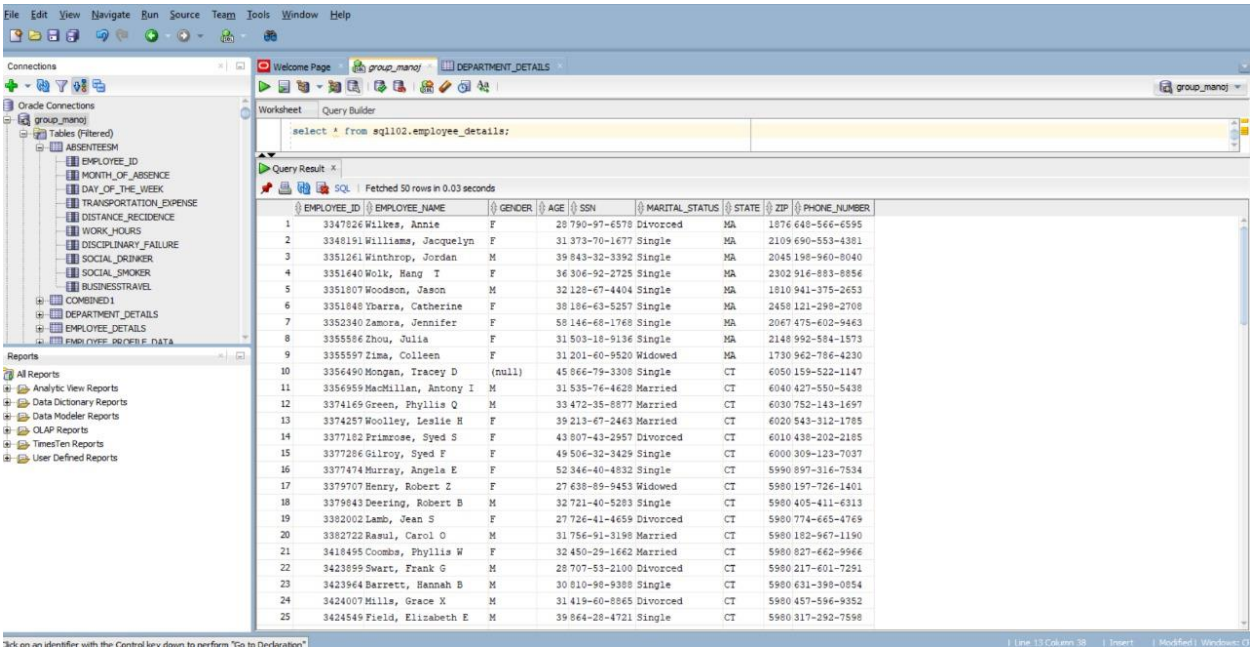
Employee Profile:

This table has Information about an employee ID, Job Id, Department ID, Location, Date of Joining and level of Education



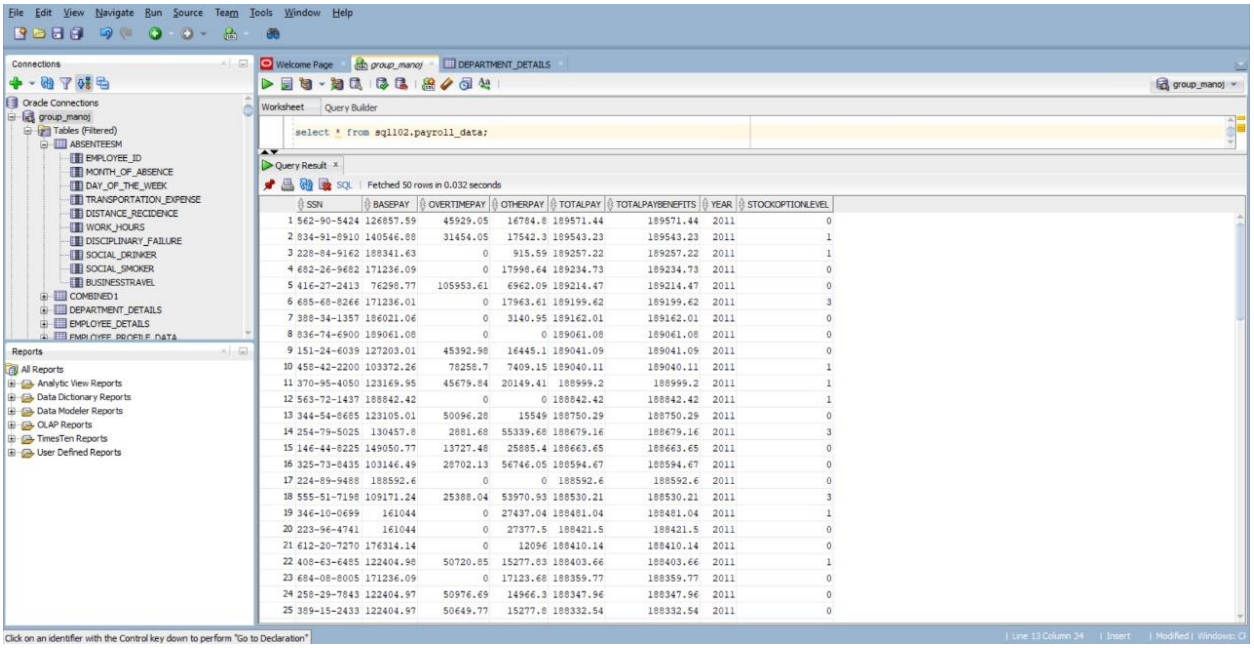
Employee Details:

This table has information about an Employee ID, First name, Last name, Age, SSN, Martial status.



Payroll Data:

This table has information about SSN, Base pay, Overtime Pay, Total Pay, Year and Benefits.

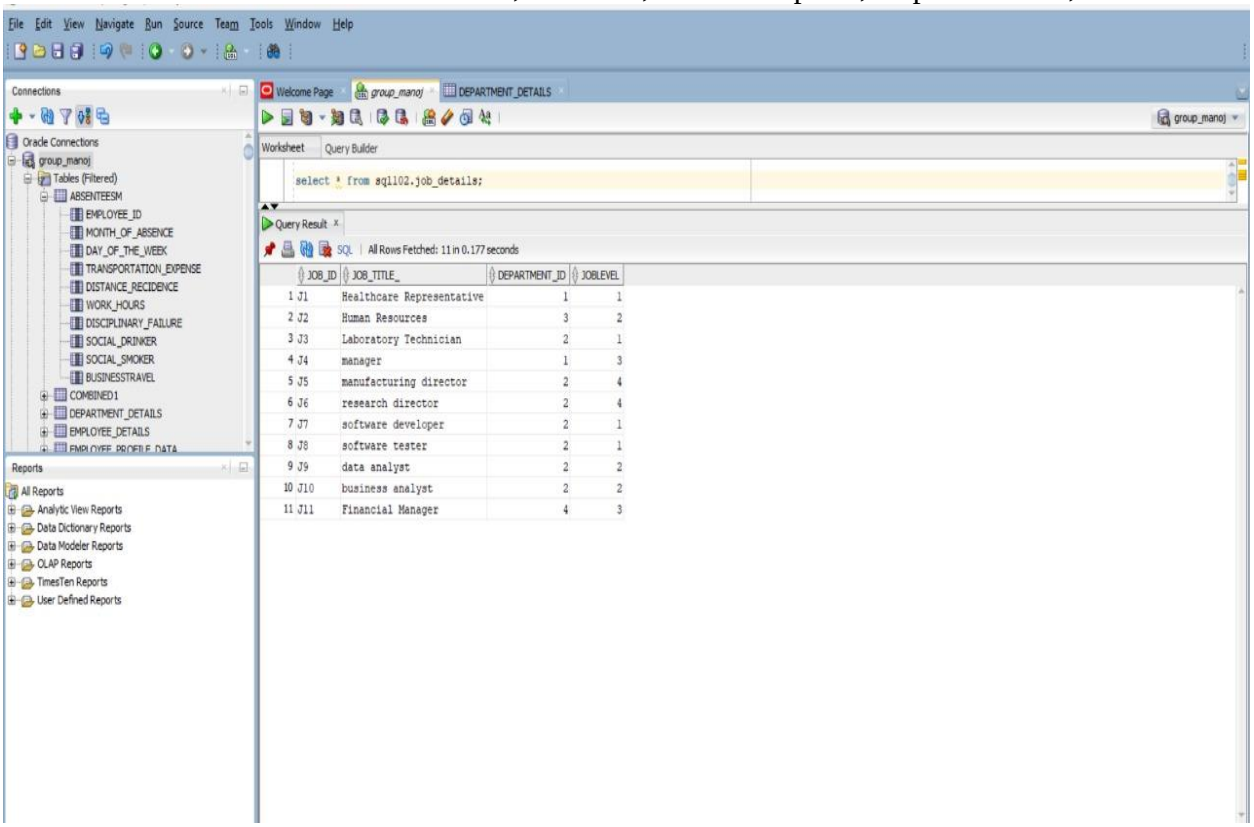


```
select * from sql102.payroll_data;
```

	SSN	BASEPAY	OVERTIMEPAY	OTHERPAY	TOTALPAY	TOTALPAYBENEFITS	YEAR	STOCKOPTIONLEVEL
1	562-90-5424	126857.59	45929.05	16784.8	189571.44	189571.44	2011	0
2	834-91-8910	140546.88	31454.05	17542.3	189543.23	189543.23	2011	1
3	228-84-9162	188341.63	0	915.59	189257.22	189257.22	2011	1
4	682-26-9682	171236.09	0	17998.64	189234.73	189234.73	2011	0
5	416-27-2413	76298.77	105953.61	6962.09	189214.47	189214.47	2011	0
6	685-68-8266	171236.01	0	17963.61	189199.62	189199.62	2011	3
7	388-34-1357	186021.06	0	3140.95	189162.01	189162.01	2011	0
8	836-74-6900	189061.08	0	0	189061.08	189061.08	2011	0
9	151-24-6039	127203.01	45392.98	16445.1	189041.09	189041.09	2011	0
10	458-42-2200	103372.26	78258.7	7409.15	189040.11	189040.11	2011	1
11	370-95-4050	123169.95	45679.84	20149.41	188999.2	188999.2	2011	1
12	563-72-1437	188842.42	0	0	188842.42	188842.42	2011	1
13	344-54-8685	123105.01	50096.28	15549	188750.29	188750.29	2011	0
14	254-79-5025	130457.8	2881.68	55339.68	188679.16	188679.16	2011	3
15	146-44-8225	149050.77	13727.48	25885.4	188663.65	188663.65	2011	0
16	325-73-8435	103146.49	28702.13	56746.05	188594.67	188594.67	2011	0
17	224-89-9488	188592.6	0	0	188592.6	188592.6	2011	0
18	555-51-7198	109171.24	25388.04	53970.93	188530.21	188530.21	2011	3
19	346-10-0699	161044	0	27437.04	188481.04	188481.04	2011	1
20	223-96-4741	161044	0	27377.5	188421.5	188421.5	2011	0
21	612-20-7270	176314.14	0	12096	188410.14	188410.14	2011	0
22	408-63-6485	122404.98	50720.85	15277.83	188403.66	188403.66	2011	1
23	684-08-8005	171236.09	0	17123.68	188359.77	188359.77	2011	0
24	258-29-7843	122404.97	50976.69	14966.3	188347.96	188347.96	2011	0
25	389-15-2433	122404.97	50649.77	15277.8	188332.54	188332.54	2011	0

Job Details:

This table has information about Job ID, Job Title, Job Description, Department ID, Job Level.

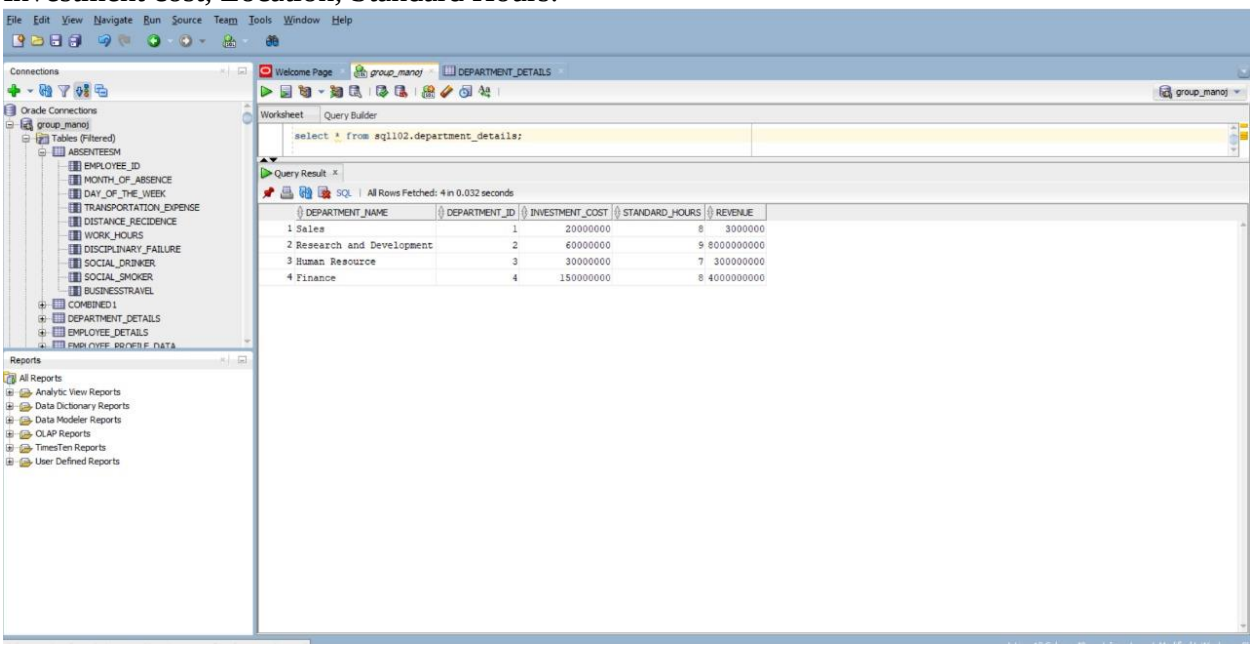


```
select * from sql102.job_details;
```

	JOB_ID	JOB_TITLE_	DEPARTMENT_ID	JOBLEVEL
1	J1	Healthcare Representative	1	1
2	J2	Human Resources	3	2
3	J3	Laboratory Technician	2	1
4	J4	manager	1	3
5	J5	manufacturing director	2	4
6	J6	research director	2	4
7	J7	software developer	2	1
8	J8	software tester	2	1
9	J9	data analyst	2	2
10	J10	business analyst	2	2
11	J11	Financial Manager	4	3

Department Details:

This table has information about the Department ID, Department name, Count, Revenue, Investment cost, Location, Standard Hours.



DATA SYNTHESIS:

The tabulation below provides a summary of the data housed in the tables,

The tabulation below provides a summary of the data housed in the tables, **Table Name**	**Columns**	**Number of constraints**	**Number of Records**
Employee Profile	7	2	1000
Employee Details	9	2	1000
Absenteesm	10	1	1000
Department Details	5	1	4
Job Details	4	1	11
Payroll data	8	1	1000

DATA INTEGRITY:

Data Integrity refers to the consistency and maintenance of the data through the life cycle of the database. In a database, data integrity can be ensured through the implementation of Integrity Constraints in a table. Integrity constraints help apply business rules to the database tables. The constraints can either be at a column level or a table level. Some of the most common constraints are,

• NOT NULL – Prevents a column from having a NULL value.
• PRIMARY KEY – Uniquely identifies each row or record in table.
• FOREIGN KEY – Uniquely identifies a column that references a PRIMARY KEY in another table.
• UNIQUE – Prevents a column from having duplicate values.
• CHECK – Checks for values that satisfy a specific condition as defined by the user.

CREATE TABLE,ALTER TABLE,INDEX :

Listed below are the constraints that were created for our database development project along with their purpose-

```
CREATE TABLE employee_profile (
 employee_id NUMBER(25) NOT NULL,
 department_id NUMBER(25) NOT NULL,
 job_id NUMBER(25) NOT NULL,
 location VARCHAR2(50 BYTE) NOT NULL,
 doj DATE NOT NULL,
education VARCHAR2(50 BYTE) NOT NULL,
education level VARCHAR2(50 BYTE) NOT NULL,
manager_id NUMBER(25) NOT NULL
)
```

CREATE UNIQUE INDEX sql102. employee_profile_pk ON

```
sql102. employee_profile (
Employee id
ASC )
TABLESPACE students PCTFREE 10
STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS
2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT )
LOGGING;
```

ALTER TABLE sql102. employee_profile

ADD CONSTRAINT **employee_profile_pk** PRIMARY

```
CREATE TABLE employee_satisfactio_rating (
 employee_id NUMBER(25) NOT NULL,
 manager_id NUMBER(25) NOT NULL,
 job_satisfaction VARCHAR2(50 BYTE) NOT NULL,
```

```
 performance_rating VARCHAR2(50 BYTE) NOT NULL,
 relationhsip_satisfaction VARCHAR2(50 BYTE) NOT NULL,
 years_with_manager NUMBER(25) NOT NULL,
)

CREATE UNIQUE INDEX sql102. employee_satisfactio_rating_pk ON
sql102. employee_satisfactio_rating (
Employee id
ASC )
TABLESPACE students PCTFREE 10
STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS
2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT )
LOGGING;
ALTER TABLE sql102. employee_satisfactio_rating
ADD CONSTRAINT employee_satisfactio_rating _pk PRIMARY



CREATE TABLE absenteesm (
 employee_id NUMBER(25) NOT NULL,
month_of_absence VARCHAR2(50 BYTE) NOT NULL,
day_of_week VARCHAR2(50 BYTE) NOT NULL,
transportation_expense NUMBER(25),
distance_from_residence NUMBER(25),
work_hours NUMBER(25),
disciplinary_failure VARCHAR2(50 BYTE) NOT NULL,
social_drinker VARCHAR2(50 BYTE) NOT NULL,
social_smoker VARCHAR2(50 BYTE) NOT NULL,
reason_of_absence VARCHAR2(50 BYTE) NOT NULL,
business_travel VARCHAR2(50 BYTE) NOT NULL
)

CREATE UNIQUE INDEX sql102. absenteesm _pk ON
sql102. absenteesm (
Employee id
ASC )
TABLESPACE students PCTFREE 10
STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS
2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT )
LOGGING;
ALTER TABLE sql102. absenteesm
ADD CONSTRAINT absenteesm_pk PRIMARY


CREATE TABLE employee_details (
 employee_id NUMBER(25) NOT NULL,
first_name VARCHAR2(50 BYTE) NOT NULL,
```

last_name VARCHAR2(50 BYTE) NOT NULL,
gender VARCHAR2(50 BYTE) NOT NULL,
age NUMBER(25) NOT NULL,
SSN VARCHAR2(50 BYTE) NOT NULL,
marital_status VARCHAR2(50 BYTE) NOT NULL,
address VARCHAR2(50 BYTE) NOT NULL,
state VARCHAR2(50 BYTE) NOT NULL,
country VARCHAR2(50 BYTE) NOT NULL,
phone_number VARCHAR2(50 BYTE) NOT NULL
)

CREATE UNIQUE INDEX sql102. employee_details _pk **ON**
sql102. employee_details (
Employee id
ASC)
TABLESPACE students PCTFREE 10
STORAGE (INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS
2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT)
LOGGING;
ALTER TABLE sql102. employee_details
ADD CONSTRAINT employee_details_pk PRIMARY

CREATE TABLE payroll (
SSN VARCHAR2(50 BYTE) NOT NULL,
base_pay NUMBER(25) NOT NULL ,
overtime_pay NUMBER(25) NOT NULL,
other_pay NUMBER(25) NOT NULL,
benefits VARCHAR2(50 BYTE) NOT NULL,
total_pay NUMBER(25) NOT NULL,
total_pay_benefits NUMBER(25) NOT NULL,
year DATE NOT NULL,
percentage_salary_hike NUMBER(25) NOT NULL,
stock_option_level NUMBER(25) NOT NULL
)

CREATE UNIQUE INDEX sql102. payroll _pk **ON**
sql102. payroll (
ssn
ASC)
TABLESPACE students PCTFREE 10
STORAGE (INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS
2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT)
LOGGING;
ALTER TABLE sql102. payroll
ADD CONSTRAINT payroll_pk PRIMARY

CREATE TABLE job_details(
job_id NUMBER(25) NOT NULL,
job_title VARCHAR2(50 BYTE) NOT NULL,
job_description VARCHAR2(50 BYTE) NOT NULL,
department_id NUMBER(25) NOT NULL,
job_level VARCHAR2(50 BYTE) NOT NULL
)

CREATE UNIQUE INDEX sql102. job_details _pk **ON**
sql102. job_details (
Job id
ASC)
TABLESPACE students PCTFREE 10
STORAGE (INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS
2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT)
LOGGING;
ALTER TABLE sql102. job_details
ADD CONSTRAINT job_details _pk PRIMARY

CREATE TABLE department_details(
department_id NUMBER(25) NOT NULL,
dept_name VARCHAR2(50 BYTE) NOT NULL,
count VARCHAR2(50 BYTE) NOT NULL,
revenue VARCHAR2(50 BYTE) NOT NULL,
investment_cost VARCHAR2(50 BYTE) NOT NULL,
location VARCHAR2(50 BYTE) NOT NULL,
standard_hours VARCHAR2(50 BYTE) NOT NULL
)

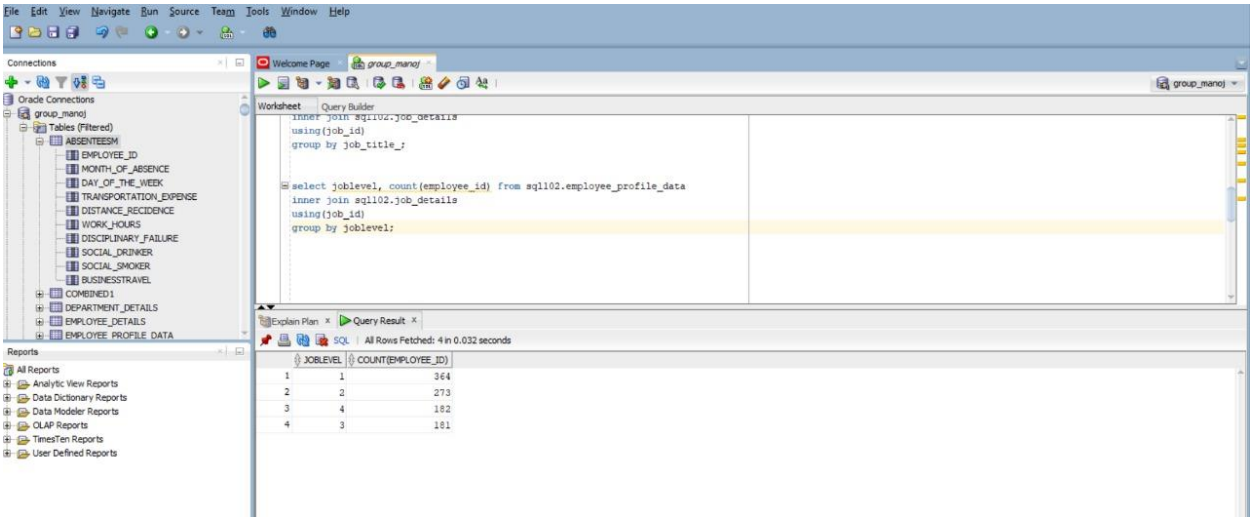
CREATE UNIQUE INDEX sql102. department_details_pk **ON**
sql102. department_details (
department_id ASC)
TABLESPACE students PCTFREE 10
STORAGE (INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS
2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT)
LOGGING;
ALTER TABLE sql102. department_details
ADD CONSTRAINT department_details_pk PRIMARY

Summary Report:

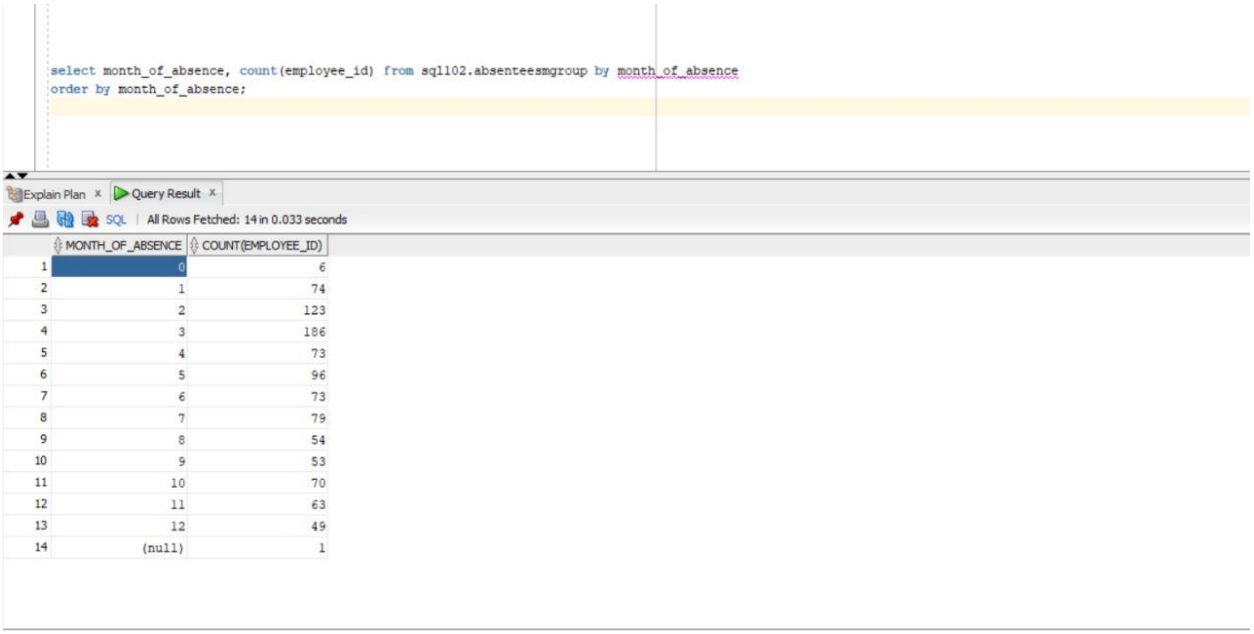
```
-- Oracle SQL Developer Data Modeler Summary Report:
--
-- CREATE TABLE                                 6
-- CREATE INDEX                                 6
-- ALTER TABLE                                  6
-- CREATE VIEW                                  0
-- ALTER VIEW                                   0
-- CREATE PACKAGE                               0
-- CREATE PACKAGE BODY                          0
-- CREATE PROCEDURE                             0
-- CREATE FUNCTION                              0
-- CREATE TRIGGER                               0
-- ALTER TRIGGER                                0
-- CREATE COLLECTION TYPE                       0
-- CREATE STRUCTURED TYPE                       0
-- CREATE STRUCTURED TYPE BODY                  0
-- CREATE CLUSTER                               0
-- CREATE CONTEXT                               0
-- CREATE DATABASE                              0
-- CREATE DIMENSION                             0
-- CREATE DIRECTORY                             0
-- CREATE DISK GROUP                            0
-- CREATE ROLE                                  0
-- CREATE ROLLBACK SEGMENT                      0
-- CREATE SEQUENCE                              0
-- CREATE MATERIALIZED VIEW                     0
-- CREATE MATERIALIZED VIEW LOG                 0
-- CREATE SYNONYM                               0
-- CREATE TABLESPACE                            1
-- CREATE USER                                  1
--
-- DROP TABLESPACE                              0
-- DROP DATABASE                                0
--
-- REDACTION POLICY                             0
-- TSDP POLICY                                  0
--
-- ORDS DROP SCHEMA                             0
-- ORDS ENABLE SCHEMA                           0
-- ORDS ENABLE OBJECT                           0
--
-- ERRORS                                       0
-- WARNINGS                                     1
```

Query Writing:

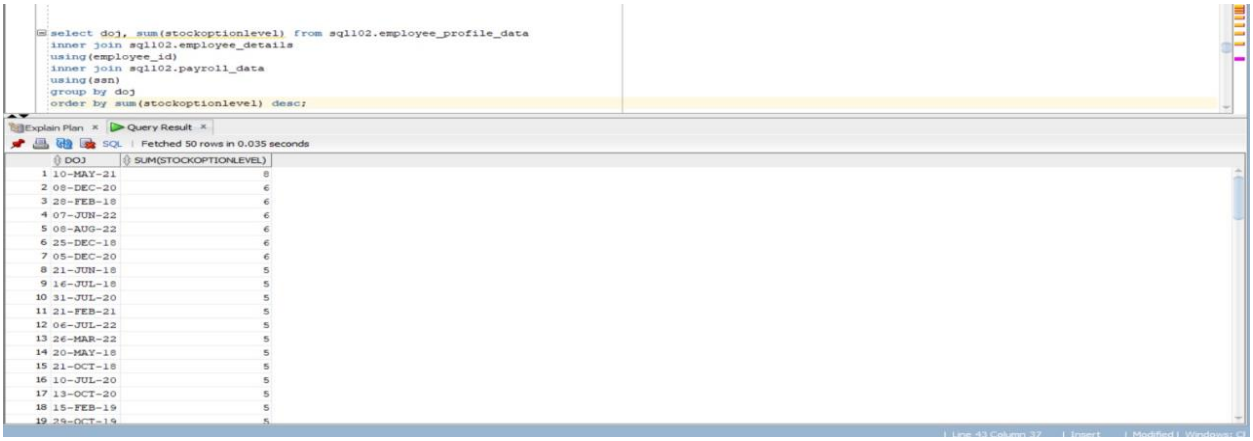
1. Fetch the list of Job ID’s present and count of Employees in it.



2.Fetch the Number of Employees absent that month.



3.Fetch the list of doj to the Stockoptionlevel.

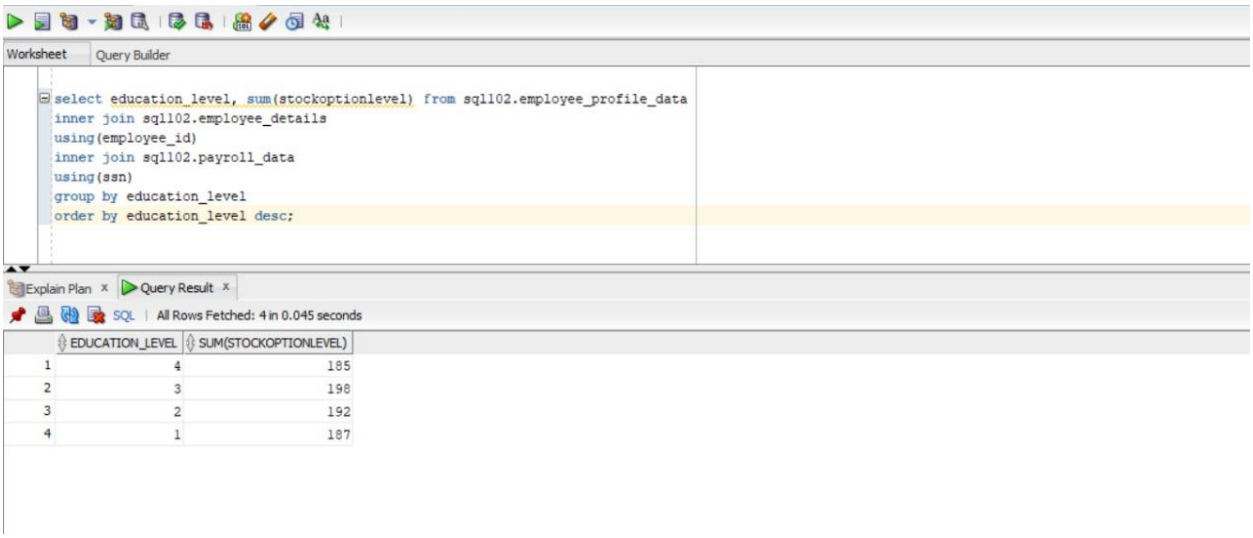


```
select doj, sum(stockoptionlevel) from sql102.employee_profile_data
inner join sql102.employee_details
using(employee_id)
inner join sql102.payroll_data
using(ssn)
group by doj
order by sum(stockoptionlevel) desc;
```

Explain Plan | Query Result | SQL | Fetched 50 rows in 0.035 seconds

	DOJ	SUM(STOCKOPTIONLEVEL)
1	10-MAY-21	8
2	08-DEC-20	6
3	28-FEB-18	6
4	07-JUN-22	6
5	08-AUG-22	6
6	25-DEC-18	6
7	05-DEC-20	6
8	21-JUN-18	5
9	16-JUL-18	5
10	31-JUL-20	5
11	21-FEB-21	5
12	06-JUL-22	5
13	26-MAR-22	5
14	20-MAY-18	5
15	21-OCT-18	5
16	10-JUL-20	5
17	13-OCT-20	5
18	15-FEB-19	5
19	29-OCT-19	5

4.Fetch the List of Education level to the stockoptionlevel.



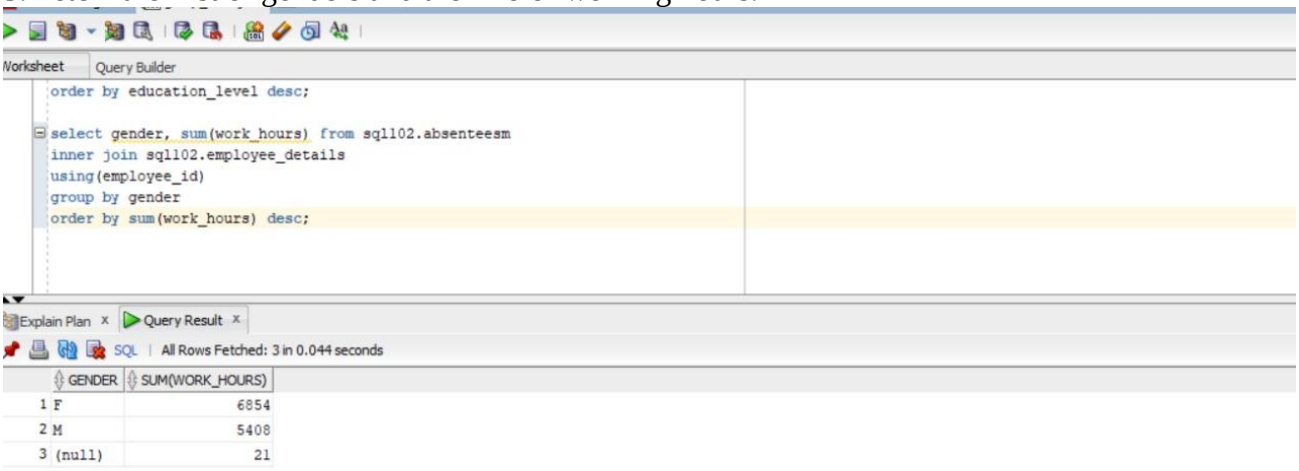
Worksheet | Query Builder

```
select education_level, sum(stockoptionlevel) from sql102.employee_profile_data
inner join sql102.employee_details
using(employee_id)
inner join sql102.payroll_data
using(ssn)
group by education_level
order by education_level desc;
```

Explain Plan | Query Result | SQL | All Rows Fetched: 4 in 0.045 seconds

	EDUCATION_LEVEL	SUM(STOCKOPTIONLEVEL)
1	4	185
2	3	198
3	2	192
4	1	187

5. Fetch the List of genders and their no of working hours.



Worksheet | Query Builder

```
order by education_level desc;

select gender, sum(work_hours) from sql102.absenteesm
inner join sql102.employee_details
using(employee_id)
group by gender
order by sum(work_hours) desc;
```

Explain Plan | Query Result | SQL | All Rows Fetched: 3 in 0.044 seconds

	GENDER	SUM(WORK_HOURS)
1	F	6854
2	M	5408
3	(null)	21

6.Fetch the List of records of disciplinary failure to the gender wise.

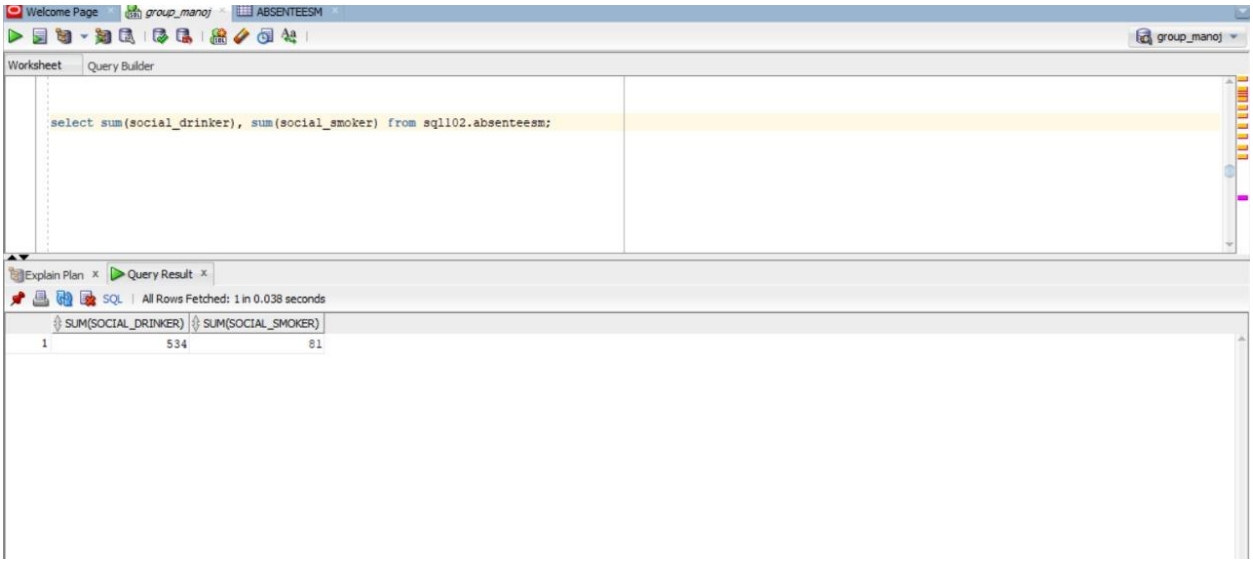
```
order by sum(work_hours) desc;


select gender, sum(disciplinary_failure) from sql102.absenteesm
inner join sql102.employee_details
using(employee_id)
group by gender
order by sum(disciplinary_failure) desc;
```

Explain Plan | Query Result | All Rows Fetched: 3 in 0.034 seconds

	GENDER	SUM(DISCIPLINARY_FAILURE)
1	F	23
2	M	19
3	(null)	1

7.Fetch the list of sum of Social drinker, Social Smoker.

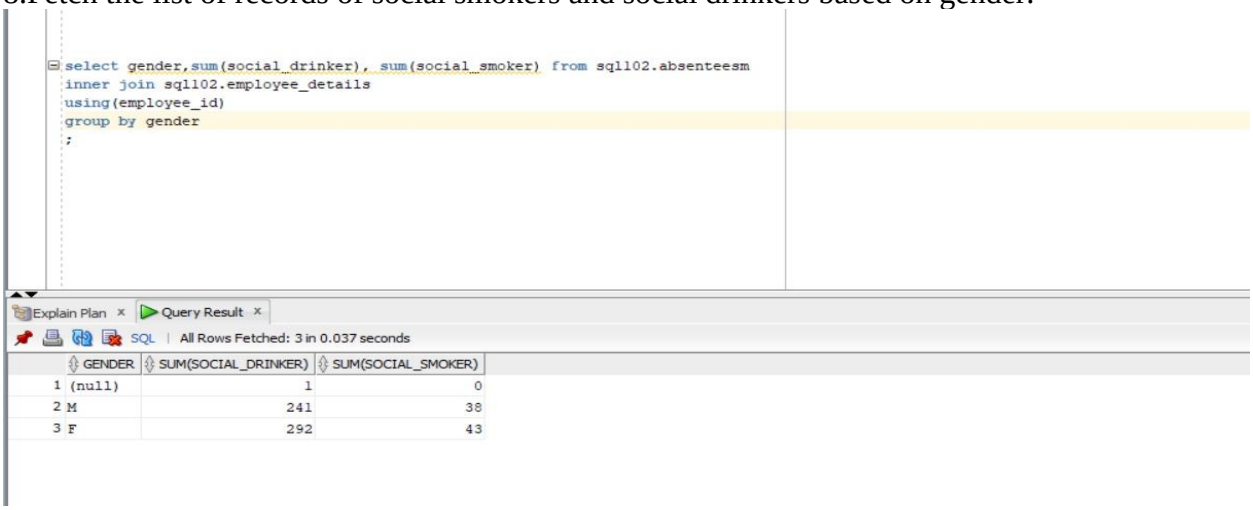


```
select sum(social_drinker), sum(social_smoker) from sql102.absenteesm;
```

Explain Plan | Query Result | All Rows Fetched: 1 in 0.038 seconds

	SUM(SOCIAL_DRINKER)	SUM(SOCIAL_SMOKER)
1	534	81

8.Fetch the list of records of social smokers and social drinkers based on gender.



```
select gender,sum(social_drinker), sum(social_smoker) from sql102.absenteesm
inner join sql102.employee_details
using(employee_id)
group by gender
;
```

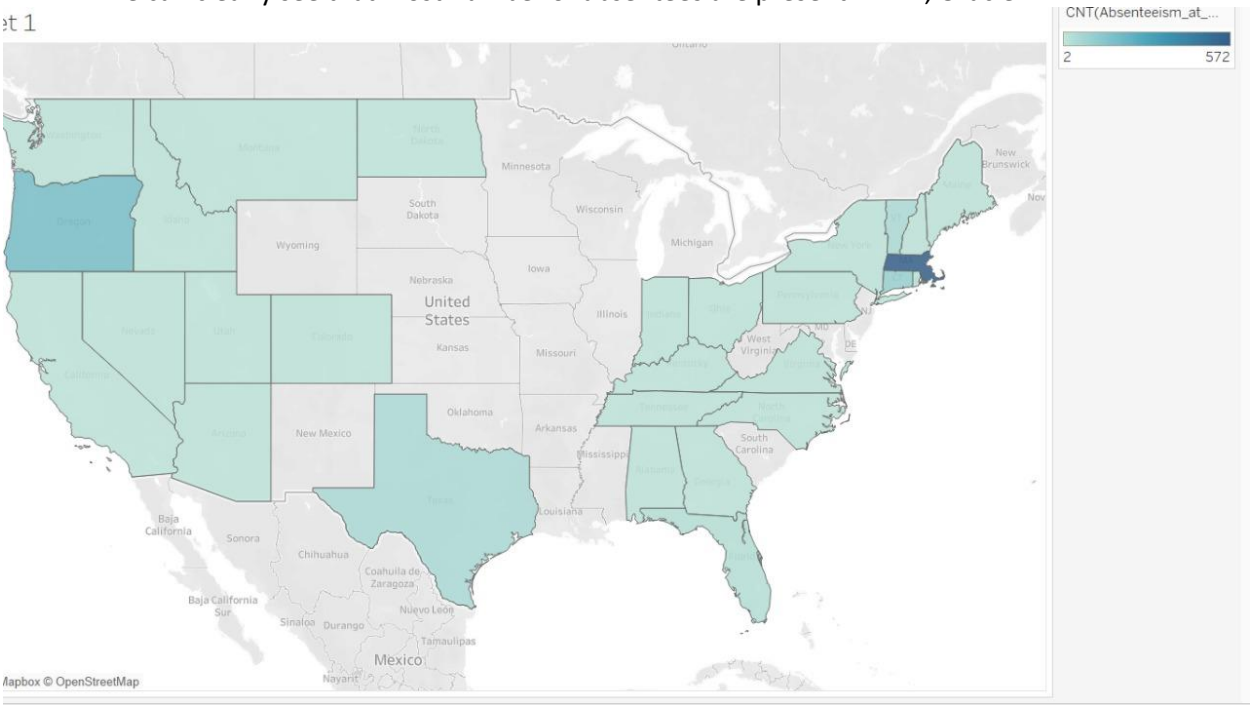
Explain Plan | Query Result | All Rows Fetched: 3 in 0.037 seconds

	GENDER	SUM(SOCIAL_DRINKER)	SUM(SOCIAL_SMOKER)
1	(null)	1	0
2	M	241	38
3	F	292	43

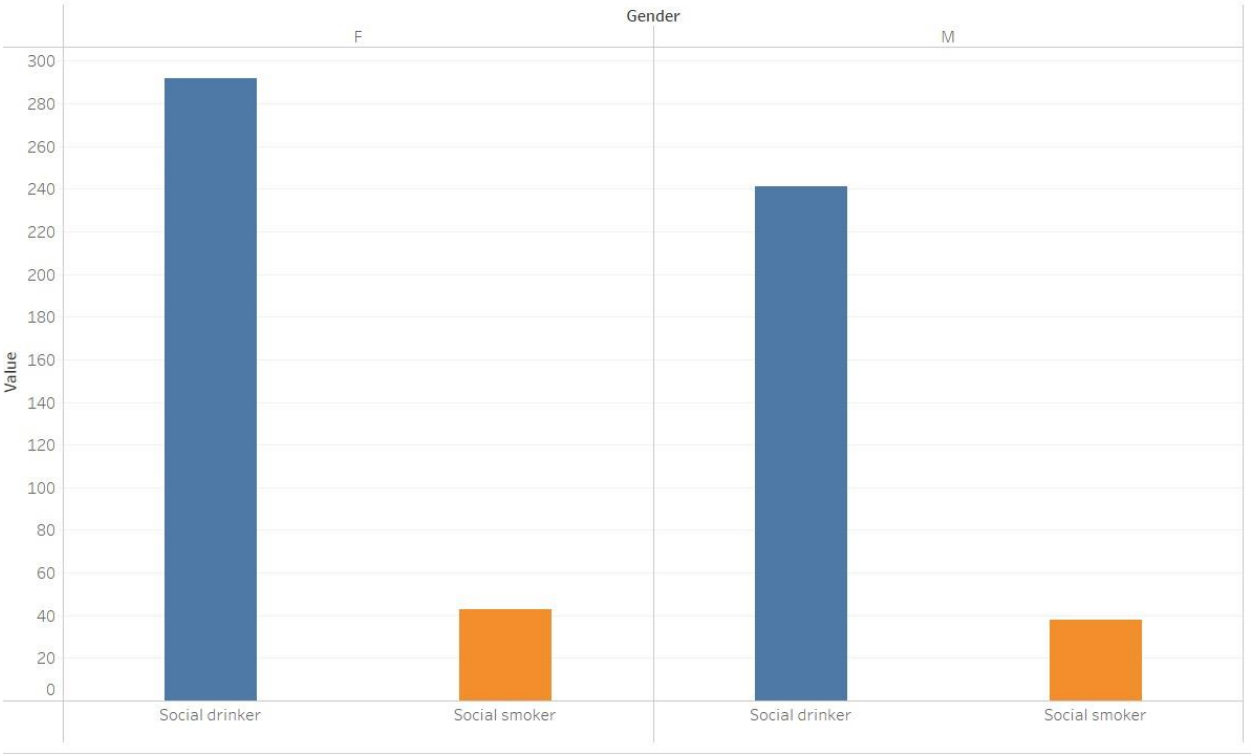
DATA VISUALIZATION

Using Tableau, we have created the following insightful Data Visualizations for our Employee database in order to show the metrics and numbers in a beautiful and clear way.

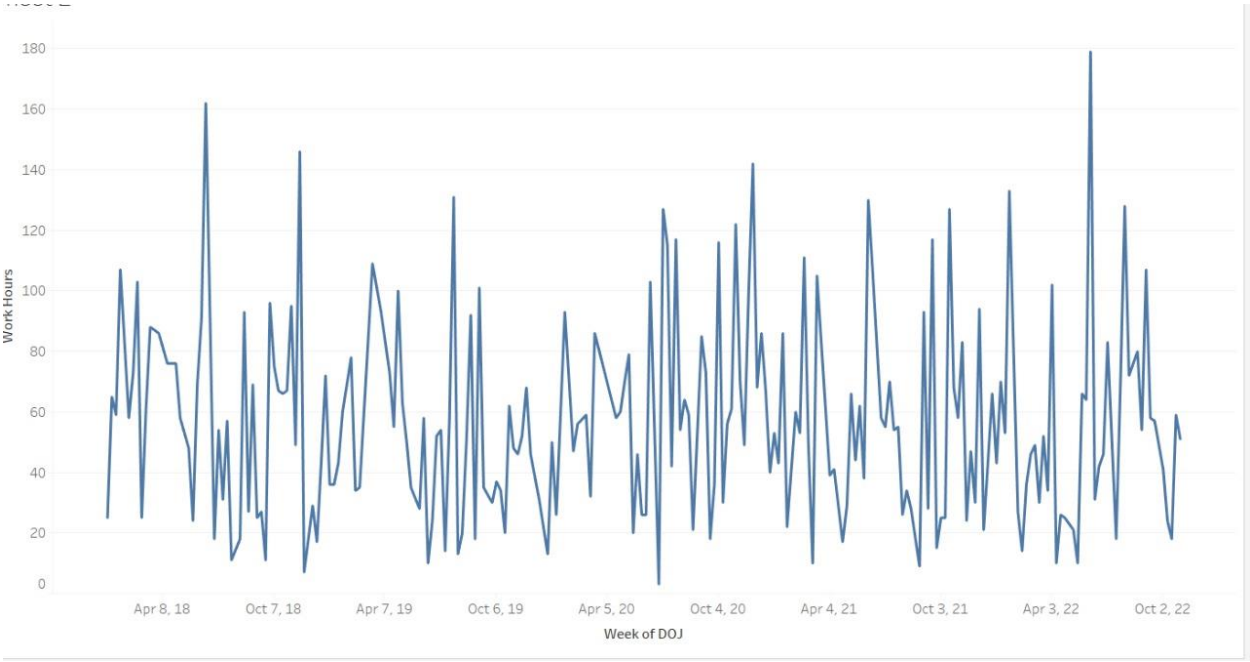
- We can clearly see that most number of absentees are present in MA, Oracle.



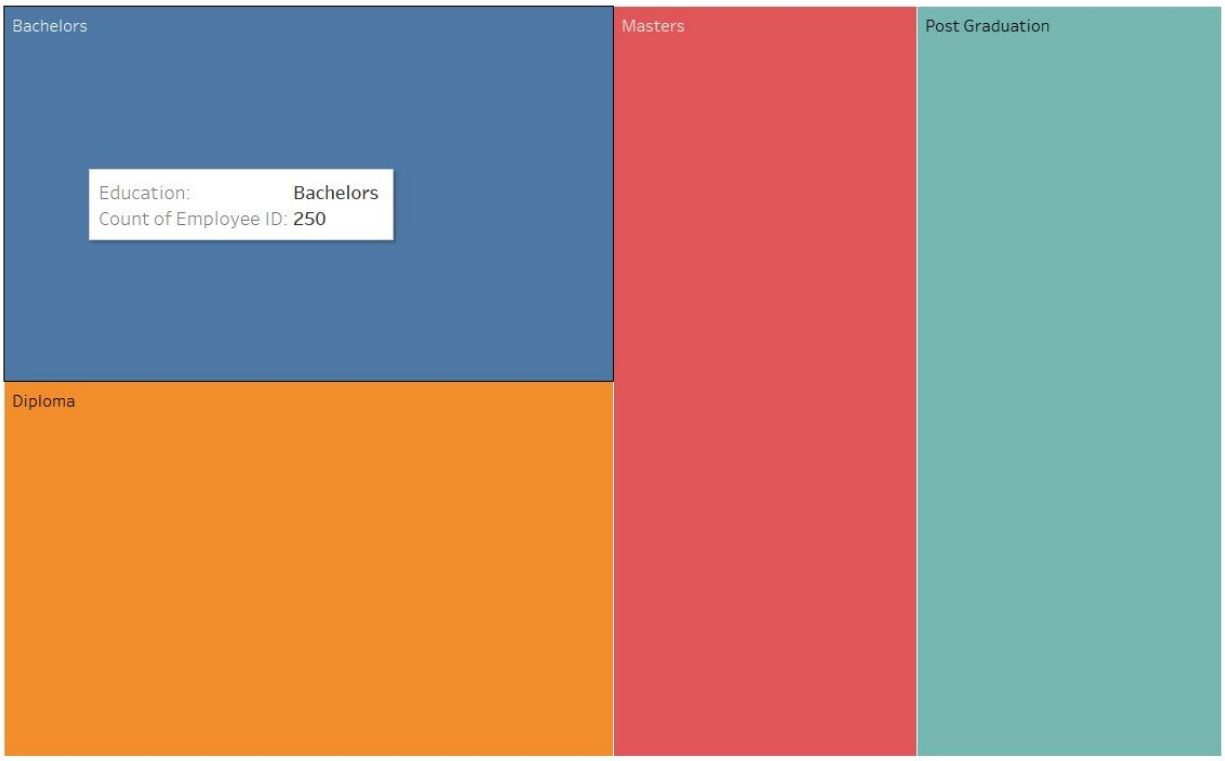
- We can Interpret that Female social drinkers and social smokers are high compared Males.



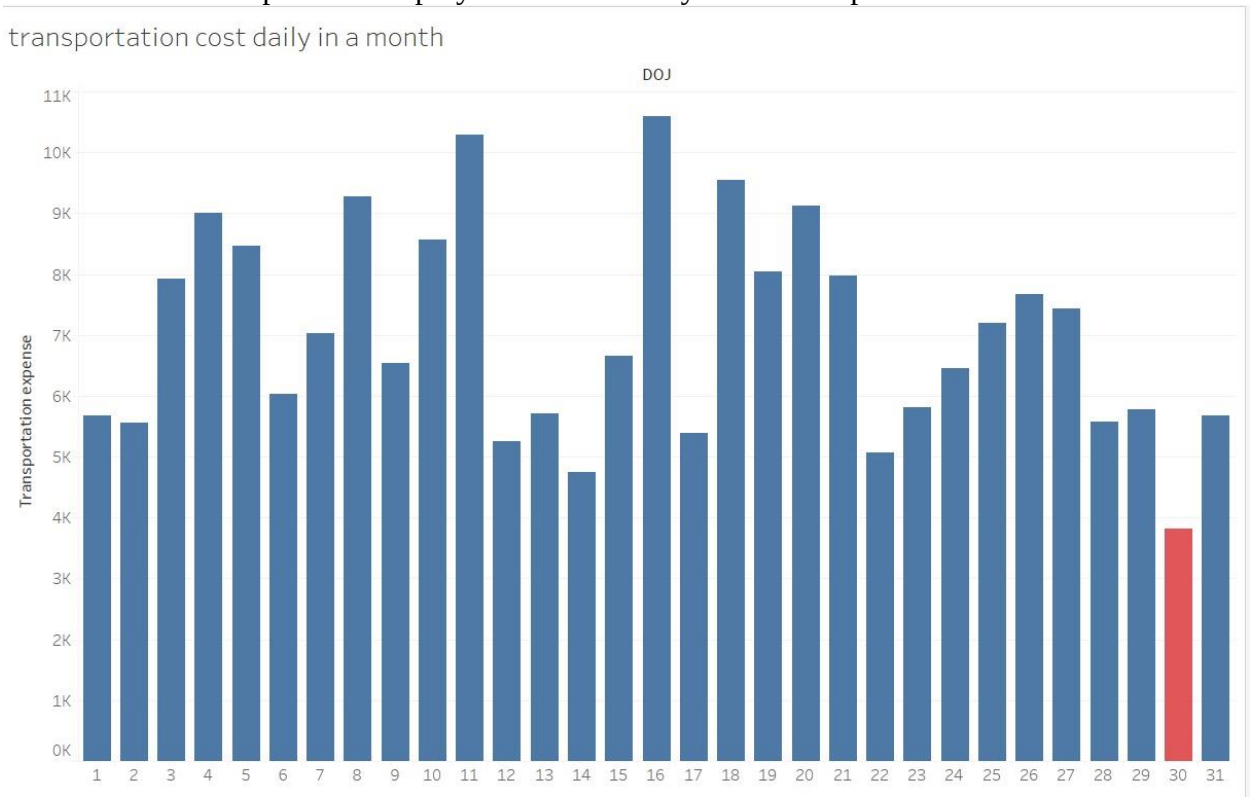
- We can see there is a fluctuation in the trend of workers vs Week of DOJ.



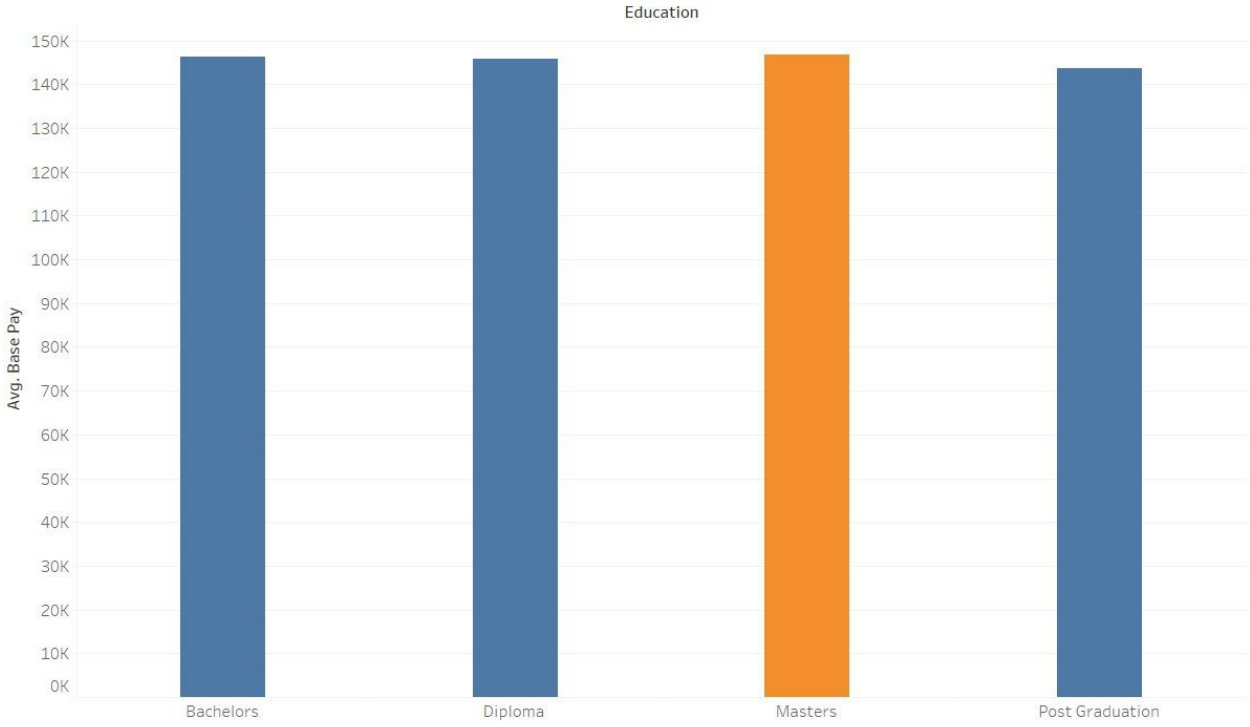
- We can interpret that most of employees working are Masters and Post-Graduation.



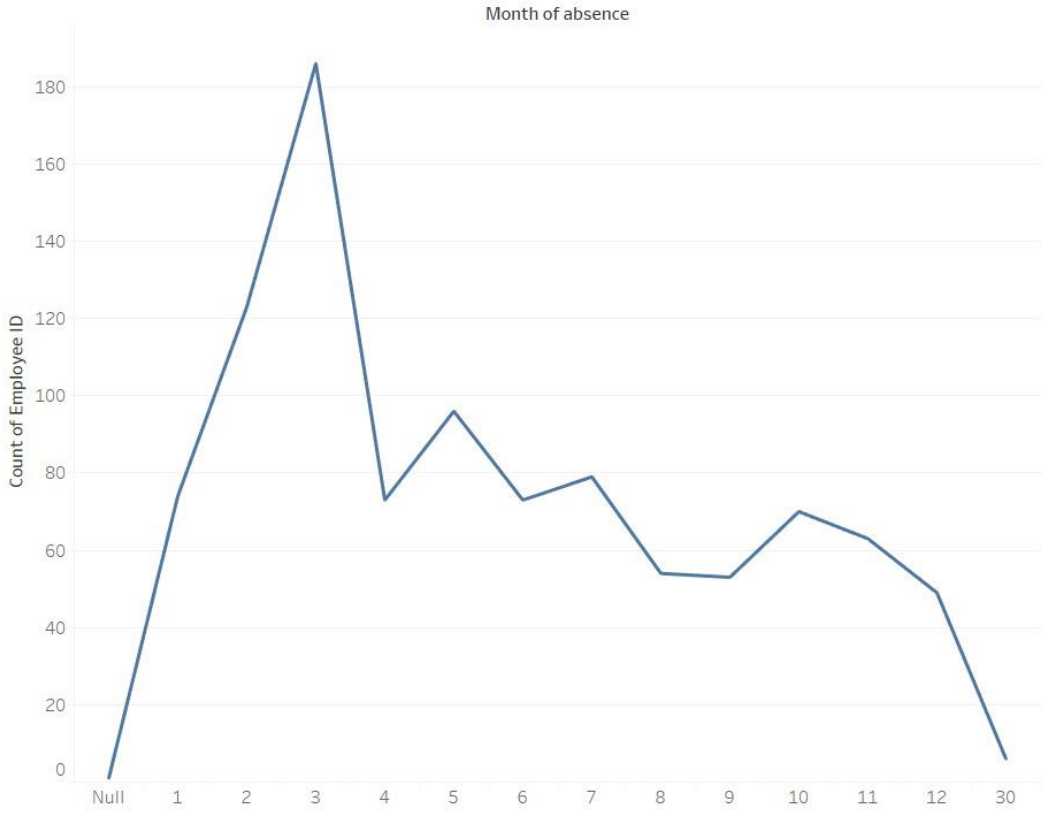
- We can interpret that employees are less likely to use transportation on 30 of the month.



- Employee pursed masters were on top of avg base pay.



- Most of the employees absentees are present in the mont of march.



- Most of the employees age ranges between 30-35