Project 1

Title: *"Employee and Department Management System"*

```
-- Create the database
CREATE DATABASE EmployeeManagement;
USE EmployeeManagement; -- Create the Department table
CREATE TABLE Departments (
DepartmentID INT PRIMARY KEY AUTO_INCREMENT,
DepartmentName VARCHAR(50) NOT NULL,
Location VARCHAR(50),
HeadOfDepartment VARCHAR(50),
AnnualBudget DECIMAL(10, 2)
); -- Create the Employee table
CREATE TABLE Employees (
EmployeeID INT PRIMARY KEY AUTO_INCREMENT,
FirstName VARCHAR(50),
LastName VARCHAR(50),
DepartmentID INT,
Salary DECIMAL(10, 2),
DateOfJoining DATE,
Email VARCHAR(100),
FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
); -- Insert records into Departments table
INSERT INTO Departments (DepartmentName, Location, HeadOfDepartment, AnnualBudget)
VALUES
('HR', 'Pune', 'Raj Sharma', 500000),
('Finance', 'Mumbai', 'Sneha Gupta', 1000000),
('IT', 'Bangalore', 'Anil Kumar', 1500000),
('Sales', 'Delhi', 'Priya Singh', 1200000),
('Operations', 'Pune', 'Ravi Patil', 900000),
('Admin', 'Chennai', 'Geeta Reddy', 400000),
('Legal', 'Hyderabad', 'Amit Desai', 600000),
('Logistics', 'Mumbai', 'Nisha Joshi', 700000),
('Research', 'Bangalore', 'Vikram Roy', 2000000),
('Support', 'Pune', 'Arun Kumar', 800000); -- Insert records into Employees table
INSERT INTO Employees (FirstName, LastName, DepartmentID, Salary, DateOfJoining,
Email)
```

```
VALUES
('Ravi', 'Sharma', 1, 45000, '2022-05-20', 'ravi.sharma@example.com'),
('Priya', 'Patil', 2, 60000, '2021-03-15', 'priya.patil@example.com'),
('Amit', 'Kumar', 3, 75000, '2020-01-10', 'amit.kumar@example.com'),
('Sneha', 'Desai', 4, 50000, '2019-07-22', 'sneha.desai@example.com'),
('Nisha', 'Joshi', 5, 55000, '2023-08-01', 'nisha.joshi@example.com'),
('Raj', 'Verma', 6, 48000, '2022-09-14', 'raj.verma@example.com'),
('Anil', 'Roy', 7, 70000, '2020-02-20', 'anil.roy@example.com'),
('Vikram', 'Chauhan', 8, 45000, '2023-06-11', 'vikram.chauhan@example.com'),
('Arun', 'Khan', 9, 65000, '2021-12-03', 'arun.khan@example.com'),
('Geeta', 'Reddy', 10, 62000, '2020-11-10', 'geeta.reddy@example.com'),
('Rohit', 'Sharma', 1, 46000, '2022-03-25', 'rohit.sharma@example.com'),
('Pooja', 'Mehta', 2, 59000, '2021-05-17', 'pooja.mehta@example.com'),
('Karan', 'Jain', 3, 73000, '2020-08-10', 'karan.jain@example.com'),
('Smita', 'Pawar', 4, 51000, '2019-09-14', 'smita.pawar@example.com'),
('Neha', 'Deshmukh', 5, 57000, '2023-10-19', 'neha.deshmukh@example.com'),
('Rahul', 'Joshi', 6, 50000, '2022-07-07', 'rahul.joshi@example.com'),
('Ajay', 'Roy', 7, 72000, '2020-03-25', 'ajay.roy@example.com'),
('Sunita', 'Sharma', 8, 48000, '2023-11-21', 'sunita.sharma@example.com'),
('Akshay', 'Patil', 9, 68000, '2021-01-15', 'akshay.patil@example.com'),
('Isha', 'Reddy', 10, 64000, '2020-06-12', 'isha.reddy@example.com'),
('Vivek', 'Shah', 1, 47000, '2022-02-23', 'vivek.shah@example.com'),
('Tina', 'Gupta', 2, 58000, '2021-06-28', 'tina.gupta@example.com'),
('Aditya', 'Kumar', 3, 72000, '2020-09-05', 'aditya.kumar@example.com'),
('Simran', 'Pawar', 4, 52000, '2019-11-08', 'simran.pawar@example.com'),
('Sanjay', 'Deshmukh', 5, 56000, '2023-05-30', 'sanjay.deshmukh@example.com'),
('Anjali', 'Joshi', 6, 51000, '2022-08-17', 'anjali.joshi@example.com'),
('Ramesh', 'Roy', 7, 74000, '2020-04-16', 'ramesh.roy@example.com'),
('Preeti', 'Sharma', 8, 49000, '2023-12-05', 'preeti.sharma@example.com'),
('Ankur', 'Patil', 9, 66000, '2021-07-13', 'ankur.patil@example.com'),
('Meera', 'Reddy', 10, 61000, '2020-10-20', 'meera.reddy@example.com'),
('Vikas', 'Shah', 1, 49000, '2022-04-04', 'vikas.shah@example.com'),
('Neeta', 'Gupta', 2, 60000, '2021-09-09', 'neeta.gupta@example.com'),
('Ashish', 'Kumar', 3, 74000, '2020-12-01', 'ashish.kumar@example.com'),
('Ritu', 'Pawar', 4, 53000, '2019-10-24', 'ritu.pawar@example.com'),
('Santosh', 'Deshmukh', 5, 58000, '2023-02-18', 'santosh.deshmukh@example.com'),
('Lata', 'Joshi', 6, 52000, '2022-11-29', 'lata.joshi@example.com'),
('Arjun', 'Roy', 7, 71000, '2020-07-04', 'arjun.roy@example.com'),
('Kiran', 'Sharma', 8, 50000, '2023-03-10', 'kiran.sharma@example.com'),
('Mohit', 'Patil', 9, 67000, '2021-11-11', 'mohit.patil@example.com'),
('Naina', 'Reddy', 10, 63000, '2020-02-14', 'naina.reddy@example.com');

select * from departments;
select * from employees;
```

1]- Situational Questions on INSERT

```
-- Situation -1
-- We have hired two employees, Siddharth Gupta (email: siddharth.gupta@example.com,
-- salary: ₹70,000, DOJ: 2024-01-01) and Aarav Jain (email: aarav.jain@example.com, salary:
-- ₹72,000, DOJ: 2024-01-02), in the IT department (DepartmentID: 3). Add these records.
insert into Employees (FirstName, LastName, DepartmentID, Salary, DateOfJoining,Email)
values('Siddharth','Gupta',3,70000, '2024-01-01','siddharth.gupta@example.com'),
('Aarav','Jain',3,72000, '2024-01-02','aarav.jain@example.com');

-- Situation -2
-- The Marketing department has been created (DepartmentID: 11) with a budget of
-- ₹13,00,000, located in Pune, and headed by Rakesh Mehta. Add this department.
insert into departments(DepartmentID, DepartmentName, Location, HeadOfDepartment,
AnnualBudget)
value(11,"Marketing",'Pune','Rakesh Mehta',1300000 );

-- Situation -3
-- Three employees have been hired for the Marketing department (DepartmentID: 11):
-- Rahul Mehra (email: rahul.mehra@example.com, salary: ₹58,000, DOJ: 2024-01-10), Sara
-- Kapoor (email: sara.kapoor@example.com, salary: ₹62,000, DOJ: 2024-01-11), and Nitin
Shah
-- (email: nitin.shah@example.com, salary: ₹60,000, DOJ: 2024-01-12). Add these records.
-- ->>
insert into Employees (FirstName, LastName, DepartmentID, Salary, DateOfJoining, Email)
value ('Rahul','Mehra',11,58000,'2024-01-10','rahul.mehra@example.com' ),
('Sara','Kapoor',11,62000,'2024-01-11','sara.kapoor@example.com'),
('Nitin','Shah',11 ,60000 ,'2024-01-12','nitin.shah@example.com');

-- Situation -4
-- Two employees have been hired for the HR department (DepartmentID: 1): Riya Sharma
-- (email: riya.sharma@example.com, salary: ₹50,000, DOJ: 2024-02-01) and Mohit Desai
(email:
-- mohit.desai@example.com, salary: ₹55,000, DOJ: 2024-02-02). Add these records.
-- ->
Insert into Employees(FirstName, LastName, DepartmentID, Salary, DateOfJoining, Email)
values ('Riya','Sharma',1 ,50000,'2024-02-01','riya.sharma@example.com'),
('Mohit','Desai',1 ,55000 ,'2024-02-24','mohit.desai@example.com');

-- Situation -5
-- Sneha Rao (email: sneha.rao@example.com) has joined the Finance department
-- (DepartmentID: 2) on 2024-02-15 with a salary of ₹60,000. Add this record.
```

-- ->
Insert into Employees(FirstName, LastName, DepartmentID, Salary, DateOfJoining, Email)
values ('Sneha','Rao',2 ,60000 ,'2024-02-15','sneha.rao@example.com');

-- Situation -6
-- Three employees have been hired for the Legal department (DepartmentID: 7): Aditya
-- Malhotra (email: aditya.malhotra@example.com, salary: ₹65,000, DOJ: 2024-03-01),
Priyanka
-- Kapoor (email: priyanka.kapoor@example.com, salary: ₹67,000, DOJ: 2024-03-02), and
Kunal
-- Singh (email: kunal.singh@example.com, salary: ₹68,000, DOJ: 2024-03-03). Add these
-- records.

Insert into Employees(FirstName, LastName, DepartmentID, Salary, DateOfJoining, Email)
values ('Aditya','Malhotra',7 ,65000,'2024-03-01','aditya.malhotra@example.com'),
('Priyanka','Kapoor',7 ,67000,'2024-02-01','priyanka.kapoor@example.com'),
('Kunal','Singh',7 ,68000 ,'2024-03-03','kunal.singh@example.com');

-- Situation -7
-- The Research department (DepartmentID: 12) has been created with a budget of
-- ₹20,00,000, located in Hyderabad, and headed by Dr. Meera Joshi. Add this department.
-- ->
Insert into departments (DepartmentID, DepartmentName, Location, HeadOfDepartment,
AnnualBudget)
value(12,"Research",'Hyderabad','Dr.Meera Joshi',2000000);

-- Situation -8
-- Two employees have been hired for the Research department (DepartmentID: 12): Arjun
-- Shah (email: arjun.shah@example.com, salary: ₹75,000, DOJ: 2024-04-01) and Riya Patel
-- (email: riya.patel@example.com, salary: ₹72,000, DOJ: 2024-04-02). Add these records.
-- ->
Insert into Employees(FirstName, LastName, DepartmentID, Salary, DateOfJoining, Email)
values ('Arjun','Shah',12 ,75000,'2024-04-01','arjun.shah@example.com'),
('Riya','Patel',12 ,72000 ,'2024-04-02','riya.patel@example.com');

-- Situation -9
-- The Logistics department (DepartmentID: 13) has been created with a budget of
-- ₹9,00,000, located in Chennai, and headed by Ravi Verma. Add this department.
-- ->
Insert into departments (DepartmentID, DepartmentName, Location, HeadOfDepartment,
AnnualBudget)
value(13,"Logistics",'Chennai','Ravi Verma',900000);

-- Situation -10
-- Two employees have been hired for the Logistics department (DepartmentID: 13): Kavita
-- Desai (email: kavita.desai@example.com, salary: ₹50,000, DOJ: 2024-05-01) and Amit Jain
-- (email: amit.jain@example.com, salary: ₹52,000, DOJ: 2024-05-02). Add these records.
-- ->
Insert into Employees(FirstName, LastName, DepartmentID, Salary, DateOfJoining, Email)
values ('Kavita','Desai',13 ,50000,'2024-05-01','kavita.desai@example.com'),
('Amit','Jain',13 ,52000 ,'2024-05-02','amit.jain@example.com');

2] Situational Questions on WHERE Clause

-- Employee Table

-- Situation -1
-- We are planning a cybersecurity project. Find all employees working in the IT
-- department (DepartmentID: 3).
-- ->
select * from employees where DepartmentID = 3;

-- Situation -2
-- To organize a corporate event, we need employees who joined after 2024-01-01.
-- Retrieve their details.
select * from employees where DateOfJoining > '2024-01-01';

-- Situation -3
-- The finance team is preparing budgets. List all employees whose salary is greater
-- than ₹60,000.
select * from employees where salary > 60000;

-- Situation -4
-- We are sending a company-wide newsletter. Find the details of employees whose
-- email ends with '@example.com'.
select * from employees where email like '%@example.com';

-- Situation -5
-- The Marketing department is launching a new campaign and needs a cost-effective
-- team. Retrieve the details of employees in the Marketing department (DepartmentID: 11)
-- who earn less than ₹60,000.
select * from employees where DepartmentID = 11 and salary < 60000;

-- Situation -6
-- A client requires a project lead for their team. Find the employee details where the
-- name starts with 'S'.
select * from employees where firstName like 's%';

-- Situation -7
-- For recruitment analysis, retrieve all employees who joined in February 2024.
select * from employees where DateOfJoining between '2024-02-01' and '2024-02-28';
select * from employees where DateOfJoining like '2024-02-__';

-- Situation -8
-- To shortlist mid-level employees, find the details of employees who are earning
-- between ₹50,000 and ₹70,000.
select * from employees where salary between 50000 and 70000;

-- Situation -9
-- We need to identify senior employees. Retrieve the details of employees who joined
-- before 2024-03-01 and earn more than ₹55,000.
select * from employees where DateOfJoining < '2024-03-01' and salary > 55000;

-- Situation -10
-- A special campaign is being planned for team leaders. Find employees who have
-- "Manager" in their name.
select * from employees where FirstName like "%manager%";

-- Department Table

-- Situation -11
-- To identify local opportunities, list all the departments located in Pune.
select * from departments where location = 'pune';

-- Situation -12
-- The board is interested in high-budget projects. Find the details of departments
-- where the budget exceeds ₹10,00,000.
select * from departments where annualbudget > 1000000;

-- Situation -13
-- For a leadership meeting, retrieve the departments headed by "Rakesh Mehta".

```
select * from departments where headofdepartment = 'rakesh mehta';

-- Situation -14
-- Marketing expansion is being planned. Find all departments whose names start
-- with "M".
select * from departments where departmentname like 'm%';


-- Situation -15
-- To allocate funds efficiently, list all departments where the budget is between
-- ₹8,00,000 and ₹15,00,000.
select * from departments where annualbudget between 800000 and 1500000;
```

3] Situational Questions on GROUP BY Clause

-- Employee Table

```
-- Situation -1:
-- To analyze salary distribution, find the total salary paid in each department.
select DepartmentID, sum(salary) "total sal" from employees
group by DepartmentID;

-- Situation -2:
-- For employee retention analysis, count the number of employees in each
-- department.
select DepartmentID, count(*) from employees
group by DepartmentID;

-- Situation -3:
-- The finance team wants to understand salary variation. Find the average salary in
-- each department.
select DepartmentID, avg(salary) "avg sal" from employees
group by DepartmentID;

-- Situation -4:
-- To reward experienced employees, identify the earliest joining date in each
-- department.
select DepartmentID, min(DateOfJoining) "earliest DOJ" from employees
group by DepartmentID;
```

```
-- Situation -5:
-- The HR team is preparing performance reports. Retrieve the maximum salary in
-- each department.
select DepartmentID, max(salary) "MAX sal" from employees
group by DepartmentID;

-- Situation -6:
-- To analyze junior-level hiring, find the minimum salary in each department.
select DepartmentID, min(salary) "min sal" from employees
group by DepartmentID;

-- Situation -7:
-- The finance team wants to review high-salary employees. Find the total salary paid
-- for employees earning more than ₹60,000 in each department.
select DepartmentID, sum(salary) from employees
where salary > 60000
group by DepartmentID;

-- Situation -8:
-- To monitor departmental growth, count the number of employees who joined in
-- 2024 in each department.
select DepartmentID, count(*) from employees
where DateOfJoining like '2024%'
group by DepartmentID;

select DepartmentID, count(*) from employees
where YEAR(DateOfJoining) = 2024
group by DepartmentID;

-- Situation -9:
-- For training allocation, count the number of employees with salaries between
-- ₹50,000 and ₹70,000 in each department.
select DepartmentID, count(*) from employees
where salary between 50000 and 70000
group by DepartmentID;

-- Situation -10: *********
--  For diversity analysis, count the number of employees whose names start with
-- each letter of the alphabet.
select upper(substring(FirstName,1,1)) as alpa_letter, count(*) from employees
group by alpa_letter
order by alpa_letter;
```

-- Department Table

```
-- Situation -11:
--  For city-based analysis, count the number of departments located in each city.
select Location, count(*) from departments
group by Location;

-- Situation -12:
-- To manage high-budget projects, find the total budget allocated to all departments
-- in each city.
select Location, sum(AnnualBudget) from departments
group by Location;

-- Situation -13:
-- For leadership tracking, count the number of departments headed by each
-- manager.
select HeadOfDepartment, count(*) from departments
group by HeadOfDepartment;

-- Situation -14:
-- To plan budget allocation, calculate the average budget of departments in each
-- city.
select Location, avg(AnnualBudget) from departments
group by Location;

-- Situation -15:
-- For funding efficiency, find the maximum and minimum budgets among
-- departments in each city.
select Location, max(AnnualBudget),min(AnnualBudget) from departments
group by Location;
```

4] Situational Questions on CONCAT and Concatenating Words

```
-- Situation -1: To create a full name column, concatenate the first name and last name of each
-- employee.
SELECT CONCAT(FirstName,' ',LastName) from employees;
```

```
-- Situation -2: For a project report, concatenate the department name and its location to display
a
-- complete department address.
SELECT CONCAT(DepartmentName,' ',Location) from departments;

-- Situation -3: To prepare a personalized email greeting, concatenate "Hello " with the
employee's
-- name and a comma.
SELECT CONCAT('Hello',', ',FirstName) from employees;

-- Situation -4: For creating a unique employee ID, concatenate the department ID and
employee
-- number.
SELECT CONCAT(EmployeeID,' ',DepartmentID) from employees;

-- Situation -5: For creating a company contact list, concatenate the employee's phone number
-- with their extension (if applicable).

-- Situation -6: To display the complete address, concatenate the street, city, and postal code of
-- each employee.

-- Situation -7: To create an employee login ID, concatenate the employee’s first name, last
name,
-- and their department ID.
SELECT CONCAT(FirstName,' ',LastName, DepartmentID) from employees;


-- Situation -8: For sending out a personalized notification, concatenate the employee's
department
-- with their position.
SELECT CONCAT(DepartmentID,' ',DepartmentName) from departments;

-- Situation -9: To display the employee's work status, concatenate their job title and current
-- project.


-- Situation -10: For an annual report, concatenate the year and employee’s first name to create
a
-- unique report reference code.
SELECT CONCAT(FirstName,' ',substring(DateOfJoining,1,4)) from employees;
```

5] Situational Questions on the UPDATE Clause

```
-- Situation -1: As part of an organizational review, we need to update the salary of all
employees
-- who are part of the IT department (DepartmentID: 3). The salary of all employees in this
-- department should be increased by 10%. Make sure that the increase is applied to every
-- employee within the department.
update employees
set salary = salary * 1.1
where departmentid = 3;

-- Situation -2: After the recent promotion evaluation, it has been decided that the employee with
-- EmployeeID 101 should be promoted to a "Senior Developer" role. Update their job title
-- accordingly and make sure the position is reflected correctly in the employee records.


-- Situation -3: The HR team has informed us of a change in the contact information of
employee
-- with EmployeeID 120. The new contact number for this employee should be updated to
-- "9876543210". Ensure that the correct phone number is reflected in the system for any future
-- communications.


-- Situation -4: The company has recently implemented a new email domain. All employees who
-- joined after January 1, 2023, need their email addresses updated to reflect the new domain
-- "@newcompany.com". Update the email addresses accordingly for all the affected
employees.
SET sql_safe_updates = 0;

SELECT employeeid, email
FROM employees
WHERE DateOfJoining > '2023-01-01';

update employees
set Email = replace(email,'@example.com',"@newcompany.com")
where DateOfJoining > '2023-01-01';

-- Situation -5: It has come to our attention that there was an error in the naming of the HR
-- department. The department name for DepartmentID 1 needs to be updated from "HR" to
-- "Human Resources" to reflect the company's official title. Perform this update to ensure
-- consistency across the records.
update departments
set DepartmentName = 'Human Resources'
```

```
where DepartmentName = 'HR';

-- Situation -6: As part of a company-wide salary review process, all employees who are
currently
-- earning below ₹50,000 need to receive a salary increase of ₹5,000. This update should only
-- affect employees with a salary below the threshold and should apply the increase uniformly to
-- each of their records.
select * from employees
where salary >= 50000;

update employees
set salary = salary + 5000
where salary < 50000;

-- Situation -7: The Marketing department has recently moved to a new office in a different city.
-- The location of the Marketing department (DepartmentID: 11) needs to be updated to reflect
this
-- change. Please update the location to "Mumbai" in the department's records accordingly.
update departments
set location = 'Mumbai'
where location = 'pune';

-- Situation -8: Update the Salary of EmployeeID 135 to ₹60,000 as part of their annual
appraisal.
update employees
set salary = 60000
where EmployeeID = 135 ;

-- Situation -9: EmployeeID 110, who was previously working on a different project, has now
-- been assigned to "Project Alpha" after the successful completion of their prior assignment.
-- Please update the project assignment for this employee to reflect this new responsibility.

-- Situation -10: The employee with EmployeeID 103 had a delayed start date due to personal
-- reasons. We need to update their joining date to the correct date, which is 2024-03-15, to
ensure
-- the records accurately reflect the employee's actual start date with the company.
update employees
set DateOfJoining = '2024-03-15'
where EmployeeID = 103 ;
```

7] Situational Questions on Table Schema Modifications (with Retrieval for Data Consistency)

-- Situation -1: Due to a change in company policy, we need to rename the Employee table to Staff.
-- Please perform the necessary operation to rename the table.
-- Note: Please do retrieve the changes made for consistency of the data. If you are changing the
-- name of the table, make sure to restore it to the original state as Employee if needed.
rename table employees to staff;
rename table staff to employees;

-- Situation -2: As the company has decided to start tracking employees' marital status, add a new
-- column MaritalStatus of type VARCHAR(20) to the Employee table.
-- Note: Please do retrieve the changes made for consistency of the data and ensure the data
-- remains accurate for all future entries.
alter table employees
add column MaritalStatus varchar(20);

-- Situation -3: The company has decided to store the employees' date of birth. Add a DateOfBirth
-- column of type DATE to the Employee table.
-- Note: Please ensure to retrieve the changes made to check the column's impact on future data
-- consistency.
alter table employees
add column DateOfBirth date;

-- Situation -4: We need to change the data type of the Salary column in the Employee table from INT
-- to DECIMAL(10,2) to accommodate fractional salary amounts.
-- Note: Please do retrieve the changes made for consistency of the data, ensuring that all existing
-- salary data is preserved accurately.
alter table employees
modify column salary decimal(10,2);

-- Situation -5: To better categorize employees, we need to add a new column EmployeeCategory in
-- the Employee table. The column should accept values like "Full-time", "Part-time", or "Contract".

```
-- Note: Please ensure the changes are reflected and retrieve the updated schema for
consistency.
alter table Employees
add column EmployeeCategory varchar(20);

-- FOR CHECKING
update Employees
set EmployeeCategory = "full-time"
where EmployeeID = 1;


-- Situation -6: The company wants to keep track of the department name for employees.
-- Therefore, a column DepartmentName should be added to the Employee table.
-- Note: After making the change, please retrieve the updates to ensure the column is populated
-- correctly for future entries.
ALTER TABLE Employees
add column DepartmentName varchar(50);

-- Situation -7: We need to set the default value of the Status column in the Employee table to
-- "Active" for any new employee added to the database.
-- Note: After applying this change, please retrieve the changes and verify that all new employee
-- records default to "Active".
alter table Employees
add column emp_Status varchar(10) default 'Active';

-- Situation -8: As part of a company-wide restructuring, the DepartmentID column in the
Employee
-- table should be renamed to TeamID to reflect the new team-based structure.
-- Note: Please retrieve the changes made for consistency of the data. If necessary, restore the
-- column name back to DepartmentID for future references.
alter table Employees
rename column DepartmentID to TeamID;
-- -
alter table Employees
rename column TeamID to DepartmentID;

-- Situation -9: The company has decided to archive old employee data, so the Employee table
needs
-- to be partitioned by JoinDate in order to keep the data for employees who joined before 2020
-- separate.  ********
-- Note: After partitioning the data, ensure to retrieve the changes and check the data
consistency
-- and accessibility for employees prior to 2020.
```

```
-- Situation -10: We need to create a foreign key constraint between the Employee table and the
-- Department table to ensure the integrity of the department data linked to each employee.
-- Note: Please retrieve the changes made to verify the foreign key relationship, ensuring data
-- integrity and consistency across both tables.
alter table Employees
add constraint emp_fk foreign key (DepartmentID) references Departments(DepartmentID);
```