1. **Business Scenario**

This HR company was developed in 2019 and has gained attention since.The company’s primary goals include providing recruiting, paperwork, human resources responsibilities, training/development, payroll processing, performance review, and compliance/legal services. In order to maintain key records and processes, we decided to utilize spreadsheets via modes like Excel. Moving forward, we hope to establish a database management system to ascertain the company's expansion.

In terms of our primitive interactions, other companies reach out to us for HR services. Thus, it is important that we prioritize processes of monitoring our company leads, hiring leads, payroll inventory, performance review information, and system security. For our company to flourish respectively, it is crucial to expand our analysis through reporting systems to better understand our hiring practices, company leads, and payroll efficiencies.

When it comes to the major channels that allow a general human resource company to function, comprehensive database management systems are essential to efficiently handle all aspects of HR management. These components entail recruiting, paperwork, training, payroll processing, performance reviews, and compliance/legal support. Understanding the advancements in this field, we understand that this system should be customizable to accommodate the unique requirements of different clients and industries, while seamlessly integrating with existing systems and software.

From our acknowledgement, a lead management module should be implemented amongst the database system to organize, trace and operate these leads appropriately. When concerning the hiring process, it should be streamlined by embodying components like job posting, candidate screening, interview scheduling, and offer management. With that, a payroll inventory module should be created to track proper employee compensation, benefits, taxes, and deductions records alike.

Aspects associated with the performance review trackings need to be integrated within the database system to handle regular employee performance evaluations. Furthermore, durable security measures should be in place to protect classified HR data that is stored in the database system. Using customizable dashboards and analytical measures will provide leighways to clients in terms of forming data-driven decisions and improving their HR functions.

When describing the database management system, it should be both scalable and flexible in nature. In doing so, it is meant to accommodate growth and aid in evolving business needs gradually throughout time. This may entail the practice of cloud-based architecture, modular design principles, and regular updates including new beneficial features and functionalities. In terms of providing proper management to the clientele, training and support should be prioritized to allow clients the ability to fully leverage the proficiency of the database management system.

Progressing into an advanced and comprehensible database management system, this HR company can better its operational efficiency, optimize client satisfaction, and place itself for continual growth and success in the competitive market of HR services.

1. **ER Diagram**



Relationship Sentences

One **Employee** can be taking many **Performance Reviews**

One **Lead Databas**e can have one to many **Hiring Database** as well as many **Employees** working at one **Lead Database**. Since One company can seek multiple job openings and many Employees can work at one job.

One **Hiring Database** can have one too many **Payroll Inventory**. Since one job can have multiple employees working in that same job.

One **Hiring Database** can have many **Performance Reviews**

One **Hiring Database** cankeepmany **Employees**

One **Payroll Inventory** can have one **employee.** Since each employee can have their own payroll information

One **Employee** could have many **Security Logs**. Since multiple logs can be associated with one employee.

**III. Conversion to Relational Model**

Lead Database (Lead ID(PK key), CompanyName(FK), ContactPerson, ContactEmail, ContactPhone, Status)

Hiring Database(JobID(PK key), PositionTitle(FK), CompanyName, JobDescription, ApplicationDeadline, CandidateName, Status)

Payroll Inventory(EmployeeID(PK key), EmployeeName, PositionTitle(FK), Department, Salary, Benefits, Taxes, Deductions, total compensation)

Performance Review(ReviewID(PK key), EmployeeID(FK), EmployeeName, ReviewDate, PerformanceRating, ManagerComments, DevelopmentPlan)

Security Logs (LogID(PK key), Date&Time, UserID(FK), Action, Description)

Employee( EmployeeID(PK key), EmployeeName(FK), PhoneNumber, Street, City, State, ZipCode, JobID )

**ER Diagram After Normalization**

****

**Relationship Sentences**

One **Company** can have many **Employers** but each employer associates with only **one Company**.

One **Company** can have many **Jobs** posted.

One **Job** can have many **Applications** from different or the same **employer**.

One **Company** can have multiple **Departments**.

Many **Employee** belongs to only one **Company**.

Many **Employees** can apply to multiple **Jobs** through multiple **Applications**.

**IV. Normalization**

**Company Table**

****

Key: CompanyID

FD1: CompanyID -Company ID, Company Name, Address, City, State, Zip

1NF: Meets the definition of a relation.

2NF: No partial Key dependencies

3NF: No Transitive dependencies.

CompanyTable (CompanyID (key), Street, City, State, ZipCode)

Key: CompanyID

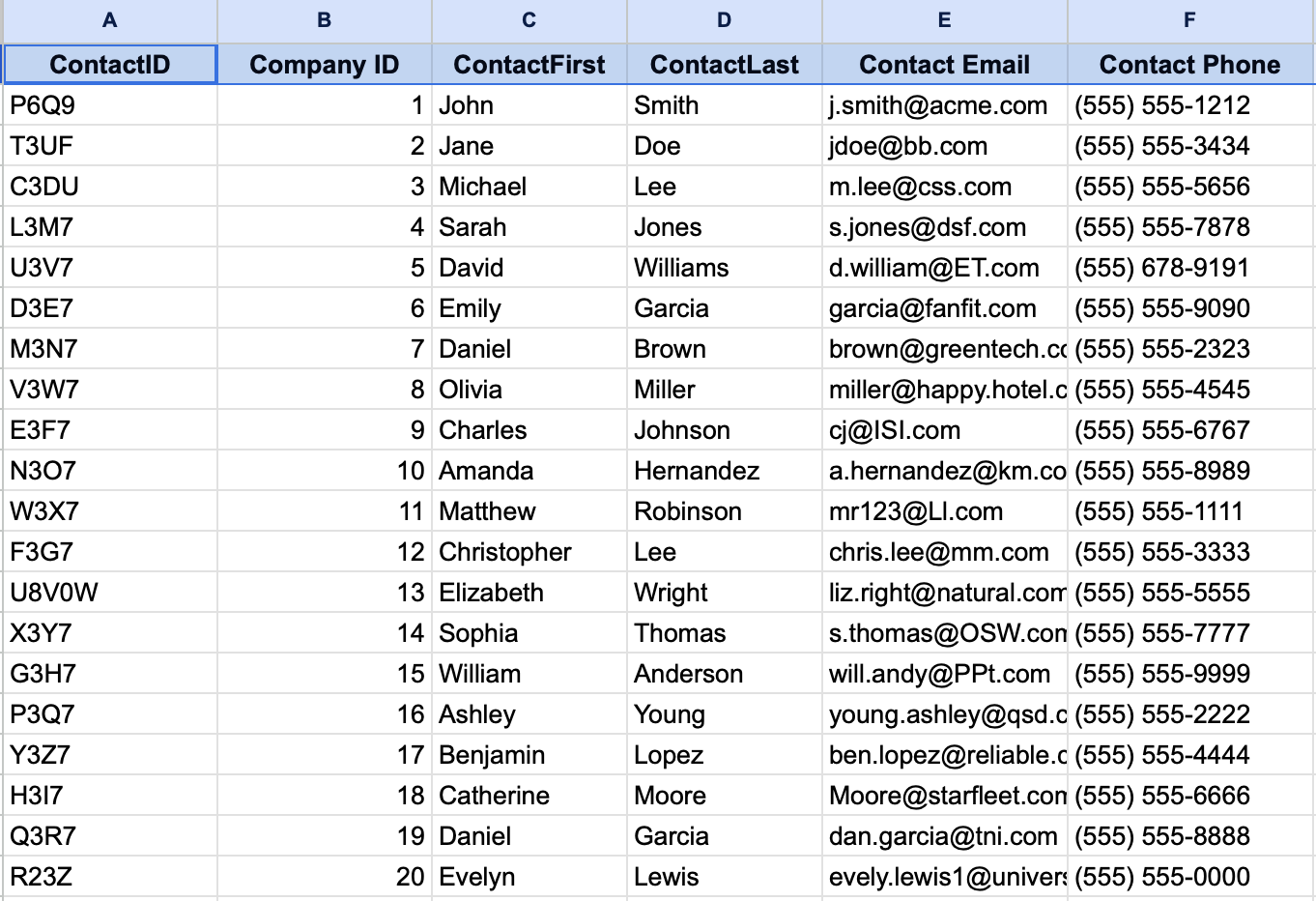
FD1: CompanyID -CompanyName, Address, City, State, ZipCode

1NF: Remove all the redundancy create separate columns for addresses and create separate columns for the contact person’s first and last name

2NF: All non-prime attributes are functionally dependent on the primary key. No partial dependencies.

3NF: Each non-prime attribute depends directly on the Company ID and has no indirect dependencies.

**Contact Table**

****

Key: Contact ID

FD1: Contact ID(PK), Company ID(FK), Contact First, Contact Last, Contact Email, Contact Phone.

FD2: CompanyID>ContactID

1NF: Separated CompanyTable into two by creating a new table to minimize redundancy.

2NF: All attributes depend on ContactID.

3NF: No transitive dependencies as every attribute directly depends on the Contact ID.

**Job Table**



Key: JobID

FD1: Job ID (Position Title, Job Description, Application Deadline)

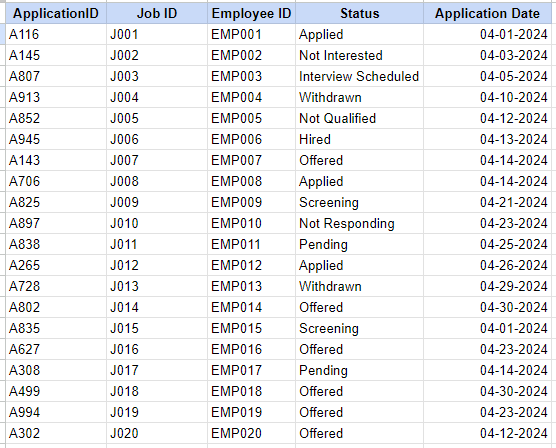
FD2:Company ID → (No other attributes are dependent on Company ID)

1NF: Attributes are atomic and there is no repeating group.

2NF: All attributes are fully functionally dependent on the Job ID. The data is 2NF.

3NF: No Transitive dependencies

**Application Table**

****

Key: ApplicationID

FD1: Application ID (Job ID, Candidate First, CandidateLas, Status)

FD2: JOB ID → (No other attributes are dependent on JobID)

1NF: Separate job information from candidate application information, as these represent different entities with distinct attributes.

2NF: All attributes are fully functionally dependent on the Job ID. The data is 2NF.

3NF: No Transitive dependencies

**Employee Table**

****

Key: EmployeeID

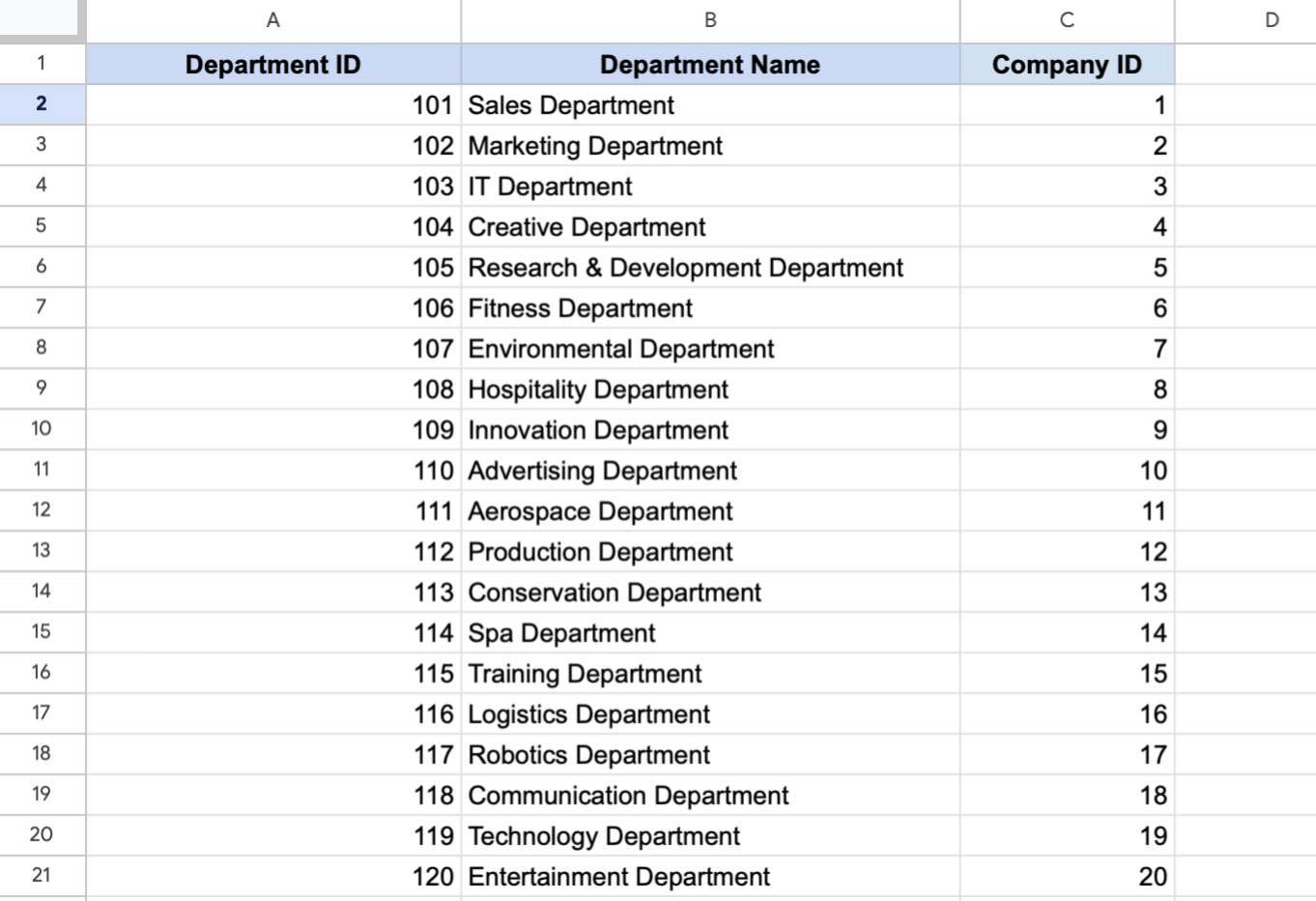
FD1: EmployeeID (FirstName, LastName, DepartmentID, Street, City, State, Zip Code, Phone Number)

FD2: Department ID → DepartmentID

1NF: Each column must contain only atomic values and each row must be unique.

2NF: All attributes are fully functionally dependent on the Employee ID.

3NF: No Transitive dependencies

**Department Table**

**Key:** DepartmentID

FD1: Department ID, Department Name, Company ID(FK)

1NF: The table contains only atomic values and there is one primary key.

2NF: There are no partial dependencies as each non-key attribute depends on the entire primary key.

3NF: No transitive dependencies

### **Final Set of Relations**

CompanyTable (CompanyID (key), CompanyName, Address, City, State, ZipCode).

ContactTable (ContactID (key), CompanyID (fk), ContactName, PhoneNumber, EmailAddress, JobTitle).

JobTable (JobID(key), CompanyID(fk), JobTitle, JobDescription, DepartmentID(fk))

Application Table ( Application ID (key), JobID, EmployeeID, Status, ApplicationDate),

Employee Table ( EmployeeID (key), FirstName, LastName, DepartmentID(fk), Street, ZipCode, PhoneNumber)

Department Table ( DepartmentID (key), Department Name, Company ID(fk) ).

V. **Creating the Database Schema with SQL (Ana Rusu)**

**Database for each of the relations in the final set of relations:**

**CREATE TABLE Company (**

**company\_id INT PRIMARY KEY,**

**company\_name VARCHAR(100),**

**address VARCHAR(200),**

**city VARCHAR(50),**

**state VARCHAR(2),**

**zip VARCHAR(10)**

**);**

**CREATE TABLE Employee (**

**employee\_id INT PRIMARY KEY,**

**first\_name VARCHAR(50),**

**last\_name VARCHAR(50),**

**position VARCHAR(100),**

**company\_id INT,**

**email VARCHAR(100),**

**phone VARCHAR(20),**

**FOREIGN KEY (company\_id) REFERENCES Company(company\_id)**

**);**

**CREATE TABLE Job (**

**job\_id INT PRIMARY KEY,**

**job\_title VARCHAR(100),**

**job\_description TEXT,**

**department\_id INT,**

**salary DECIMAL(10, 2)**

**);**

**CREATE TABLE Application (**

**application\_id INT PRIMARY KEY,**

**job\_id INT,**

**applicant\_name VARCHAR(100),**

**application\_date DATE,**

**status VARCHAR(50),**

**FOREIGN KEY (job\_id) REFERENCES Job(job\_id)**

**);**

**CREATE TABLE Employer (**

**employer\_id INT PRIMARY KEY,**

**employer\_name VARCHAR(100),**

**address VARCHAR(200),**

**phone VARCHAR(20),**

**email VARCHAR(100)**

**);**

**CREATE TABLE Department (**

**department\_id INT PRIMARY KEY,**

**department\_name VARCHAR(100),**

**employer\_id INT,**

**FOREIGN KEY (employer\_id) REFERENCES Employer(employer\_id)**

**);**

### **Adding Foreign Keys**

**ALTER TABLE Job**

**ADD CONSTRAINT FK\_Job\_Department**

**FOREIGN KEY (DepartmentName) REFERENCES Department(DepartmentName),**

**ADD CONSTRAINT FK\_Job\_Company**

**FOREIGN KEY (CompanyName) REFERENCES Company(CompanyName);**

**ALTER TABLE Employer**

**ADD CONSTRAINT FK\_Employer\_Company**

**FOREIGN KEY (CompanyName) REFERENCES Company(CompanyName);**

**ALTER TABLE Application**

**ADD CONSTRAINT FK\_Application\_Employee**

**FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),**

**ADD CONSTRAINT FK\_Application\_Job**

**FOREIGN KEY (JobTitle, CompanyName) REFERENCES Job(JobTitle, CompanyName);**

**ALTER TABLE Employee**

**ADD CONSTRAINT FK\_Employee\_Department**

**FOREIGN KEY (DepartmentName) REFERENCES Department(DepartmentName);**

**ALTER TABLE Company**

**ADD CONSTRAINT UC\_CompanyName UNIQUE (CompanyName);**

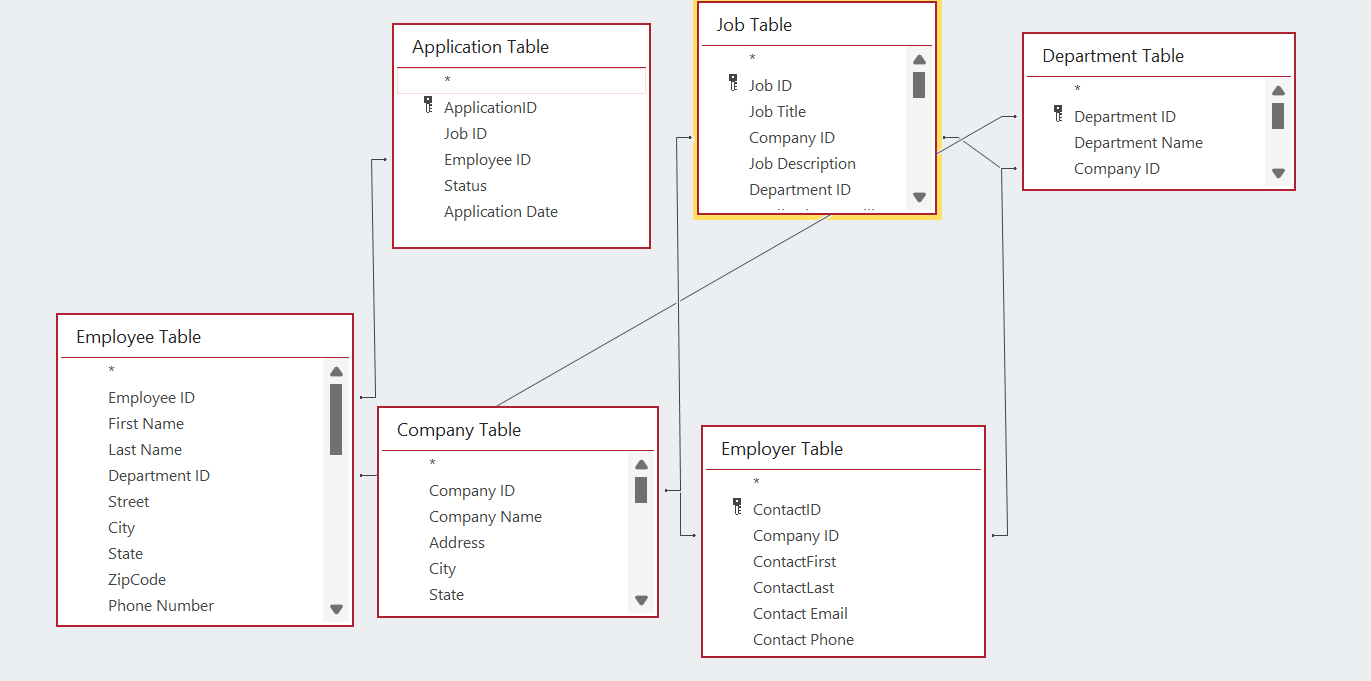
**ALTER TABLE Department**

**ADD CONSTRAINT UC\_DepartmentName UNIQUE (DepartmentName);**

**Query SQL:**

**SELECT [Employee Table].[Employee ID], [Employee Table].[First Name], [Employee Table].[Last Name], [Employee Table].[Phone Number], [Application Table].ApplicationID, [Application Table].Status, [Company Table].[Company Name], [Company Table].Address, [Employer Table].ContactFirst, [Employer Table].ContactLast, [Employer Table].[Contact Email], [Department Table].[Department Name], [Job Table].[Job Title], [Job Table].[Application Deadline]**

**FROM (([Job Table] INNER JOIN [Department Table] ON [Job Table].[Company ID] = [Department Table].[Company ID]) INNER JOIN ([Company Table] INNER JOIN [Employer Table] ON ([Company Table].[Company ID] = [Employer Table].[Company ID]) AND ([Company Table].[Company ID] = [Employer Table].[Company ID])) ON ([Job Table].[Company ID] = [Company Table].[Company ID]) AND ([Department Table].[Company ID] = [Employer Table].[Company ID])) INNER JOIN ([Application Table] INNER JOIN [Employee Table] ON [Application Table].[Employee ID] = [Employee Table].[Employee ID]) ON [Department Table].[Department ID] = [Employee Table].[Department ID];**

****

**SQL FOR INSERT:**

INSERT INTO Employee VALUES(‘EMP021’, ‘Mary’, ‘Jane’, ‘121’, ‘123 Main str’, ‘New York’, ‘NY’, ‘1001’, ‘217-987-0000’)

INSERT INTO Employee VALUES(‘EMP022’, ‘Mason’, ‘Lee’, ‘122’, ‘1 Fulton str’, ‘New York’, ‘NY’, ‘1002’, ‘217-907-6578’)

INSERT INTO Employee VALUES(‘EMP023’, ‘John’, ‘Luis’, ‘123’, ‘555 67th str’, ‘Chicago’, ‘IL’, ‘10222’, ‘917-927-0777’)

INSERT INTO Employee VALUES(‘EMP023’, ‘Kevin’, ‘Goldberg’, ‘124’, ‘67 12 ave ’, ‘New York’, ‘NY’, ‘1007’, ‘217-999-9999’)

INSERT INTO Employee VALUES(‘EMP024’, ‘Margaret’, ‘Fong’, ‘125’, ‘55 Random str’, ‘Los Angeles’, ‘CA’, ‘10786’, ‘123-987-6754’)

INSERT INTO Company VALUES(‘21’, ‘Shell’, ‘123 Love str’, ‘San Francisco’, ‘CA’, ‘12377’)

INSERT INTO Company VALUES(‘22’, ‘Disney’, ‘24 5th str’, ‘Seatle’, ‘WA’, ‘14445’)

INSERT INTO Company VALUES(‘23’, ‘NBC’, ‘145 Each str’, ‘Portland’, ‘OR’, ‘87654’)

INSERT INTO Company VALUES(‘24’, ‘Truck ’, ‘78 71st str’, ‘Boise’, ‘ID’, ‘15567’)

INSERT INTO Company VALUES(‘25’, ‘TD Bank’, ‘67 Avenue str’, ‘Sacramento’, ‘CA’, ‘19875’)

INSERT INTO Department VALUES( ‘121’, ‘Technology Department’, ‘23’)

INSERT INTO Department VALUES( ‘122’, ‘Sales Department’, ‘24’)

INSERT INTO Department VALUES( ‘123’, ‘Human Resources Department’, ‘25’)

INSERT INTO Department VALUES( ‘124’, ‘Marketing Department’, ‘26’)

INSERT INTO Department VALUES( ‘125’, ‘Management Department’, ‘27’)

INSERT INTO Job VALUES(‘J021’, ‘SDET’, ‘21’, ‘It Tests’, ‘121’, ‘2024-05-12’)

INSERT INTO Job VALUES(‘J022’, ‘HR’, ‘22’, ‘Human Resources’, ‘122’, ‘2024-03-01’)

INSERT INTO Job VALUES(‘J023’, ‘Sales’, ‘23’, ‘General Sales’, ‘123’, ‘2024-01-12’)

INSERT INTO Job VALUES(‘J024’, ‘Analyst’, ‘24’, ‘Analyze Dtat’, ‘124’, ‘2024-02-22’)

INSERT INTO Job VALUES(‘J025’, ‘Manager’, ‘25’, ‘Staf management’, ‘125’, ‘2024-01-25’)

INSERT INTO Employer VALUES(‘Y3ZHF’, ‘21’, ‘John’, ‘McDonald’, ‘mcdonald@email.com’, ‘234-876-9991’)

INSERT INTO Employer VALUES(‘DT54R’, ‘22’, ‘Mike’, ‘Fin’, ‘mfin@gmail.com’, ‘834-876-9876’)

INSERT INTO Employer VALUES(‘RT89H’, ‘23’, ‘Anna’, ‘Russo’, ‘Arusso@Yahoo.com’, ‘534-876-5555’)

INSERT INTO Employer VALUES(‘Y5HTF’, ‘24’, ‘Bob’, ‘Marley’, ‘bobM@mail.com’, ‘765-876-8754’)

INSERT INTO Employer VALUES(‘DR5Y’, ‘25’, ‘Tom’, ‘Brady’, ‘TBrady@email.com’, ‘134-76-9744’)

INSERT INTO Application VALUES(‘A765’, ‘J020’,’EMP020’, ‘Applied’, ‘03-12-2024’)

INSERT INTO Application VALUES(‘A766’, ‘J021’,’EMP021’, ‘Pendin’, ‘01-22-2024’)

INSERT INTO Application VALUES(‘A775’, ‘J022’,’EMP022’, ‘Offered’, ‘02-12-2024’)

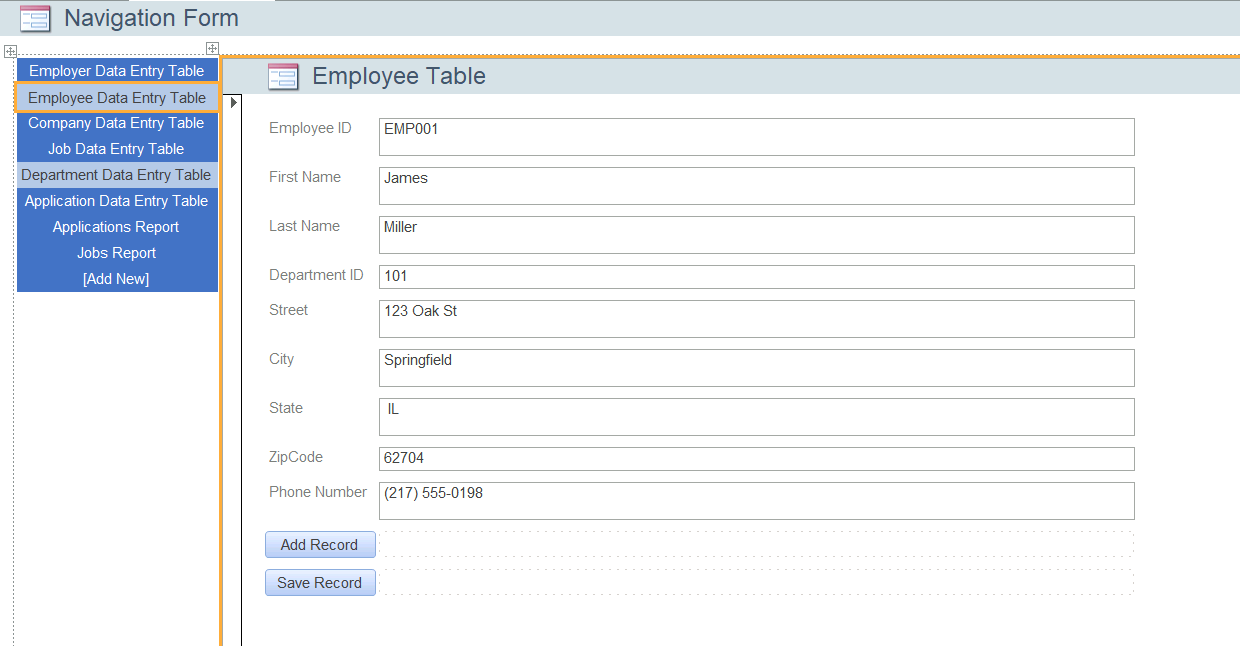
INSERT INTO Application VALUES(‘A795’, ‘J023’,’EMP023’, ‘Not interested’, ‘04-05-2024’)

INSERT INTO Application VALUES(‘A781’, ‘J024’,’EMP024’, ‘Applied’, ‘05-02-2024’)

VI. **Database Application (Edis)**

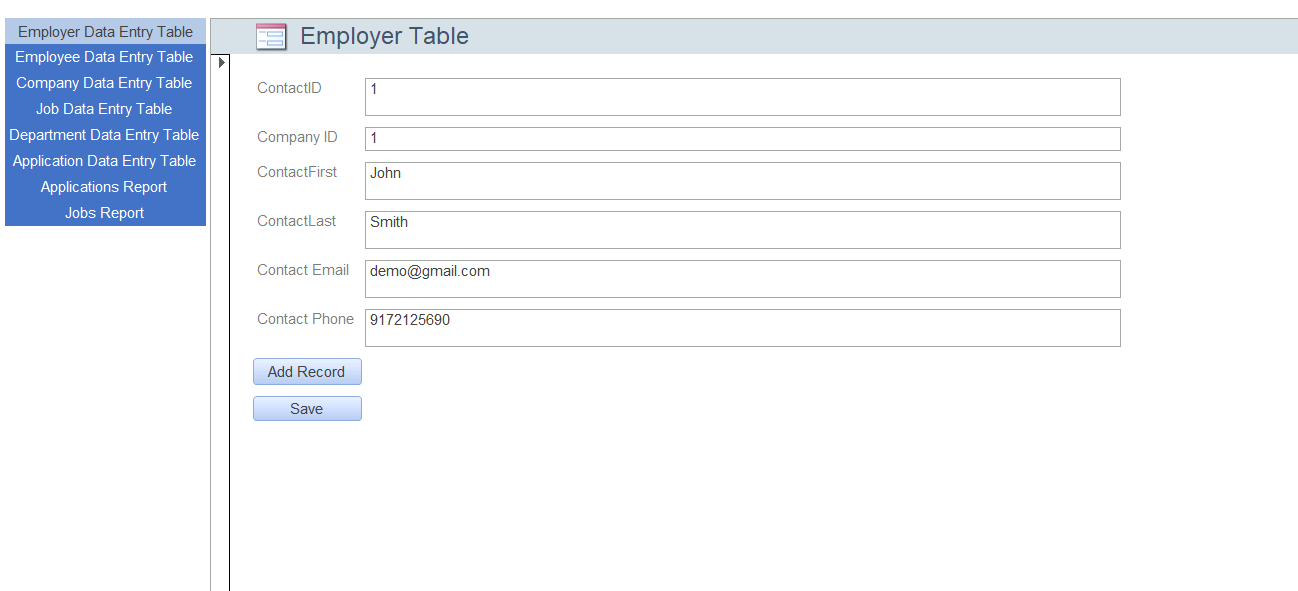
***Navigation Form***

This form will allow access and navigation to the rest of the database forms.

****

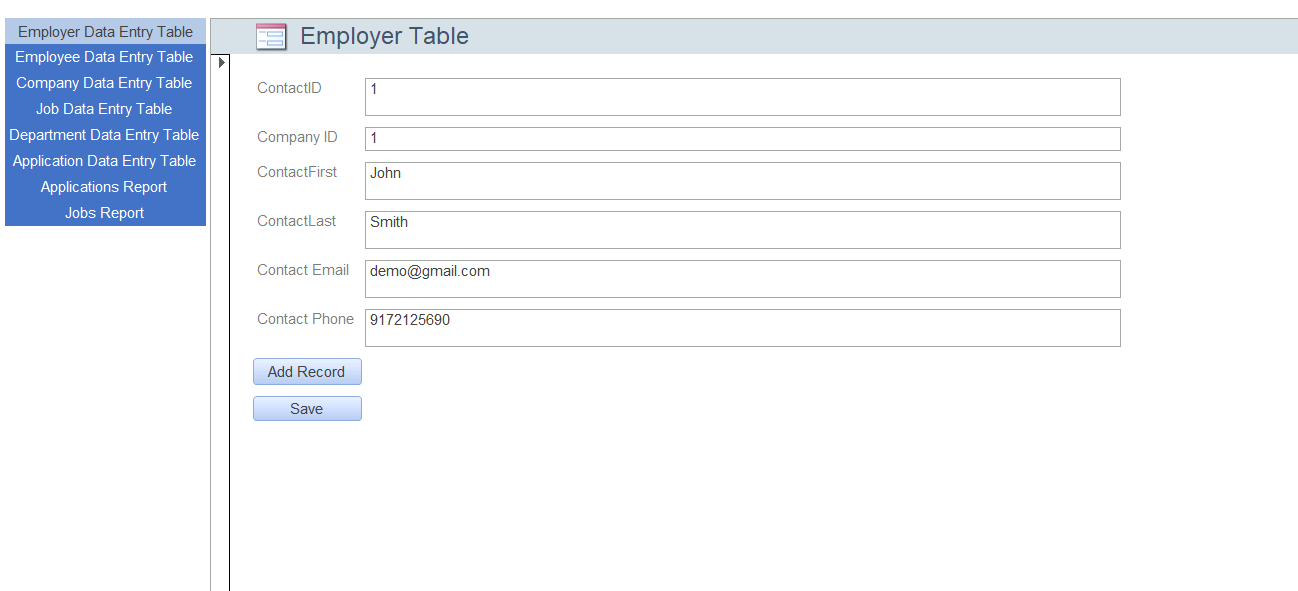
***Employer Data Entry Table***

This form will allow access to create employers and edit previous employers

****

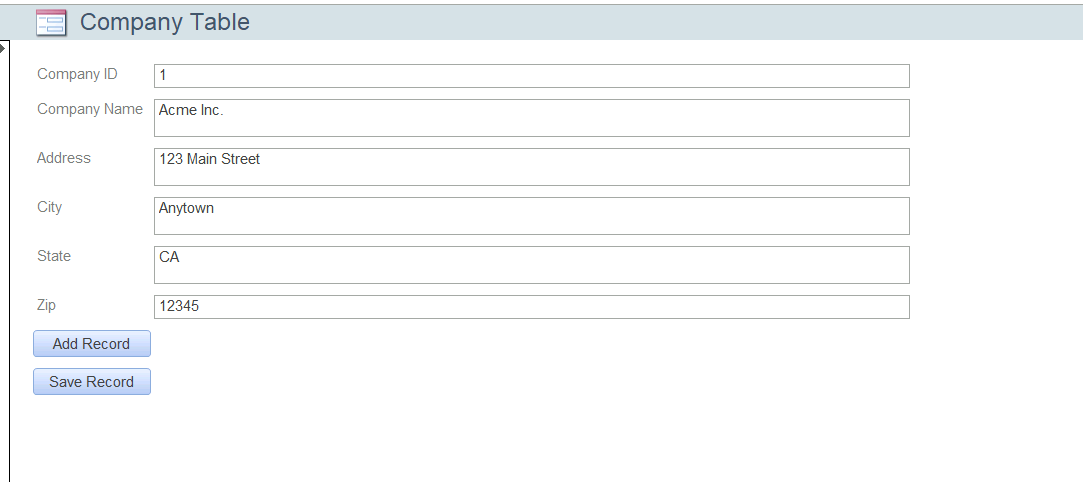
***Employee Data Entry Table***

This form will allow access to create employees and edit previous employees

****

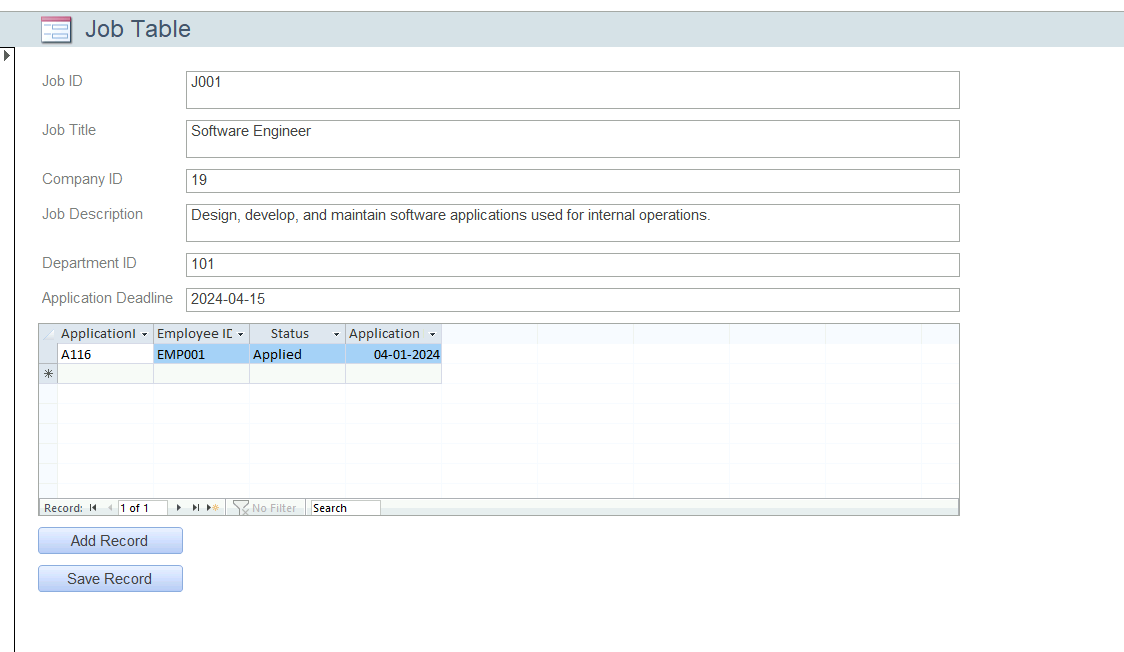
***Company Data Entry Table***

This form will allow access to create companies and edit previous companies

****

***Job Data Entry Table***

This form will allow access to create jobs and edit job companies, and view all the applications for a particular job.



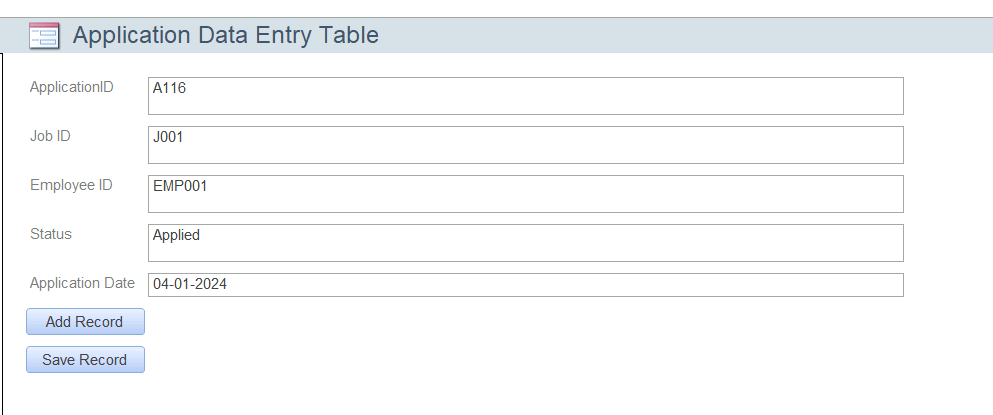
***Department Data Entry Table***

This form will allow access to create departments and edit departments. It also give an easy access to the "Create company" form, to manage companies.



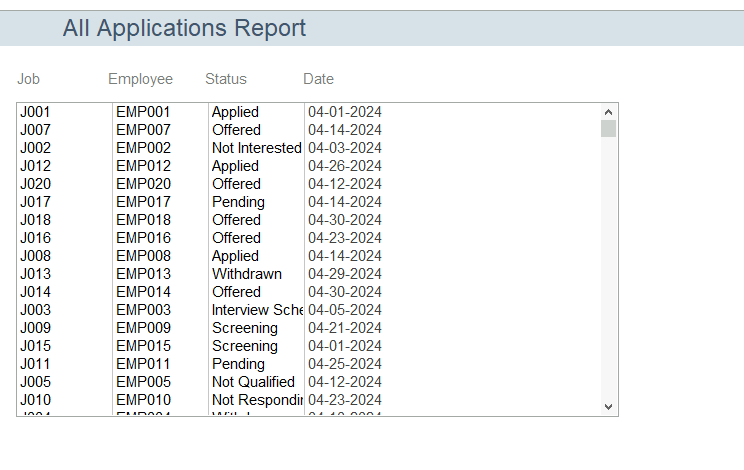
***Application Data Entry Table***

This form will allow access to create applications and edit applications (if needed).



***Application Report***

This form will allow access to see applications, with their job ID, employee ID, status, and date.

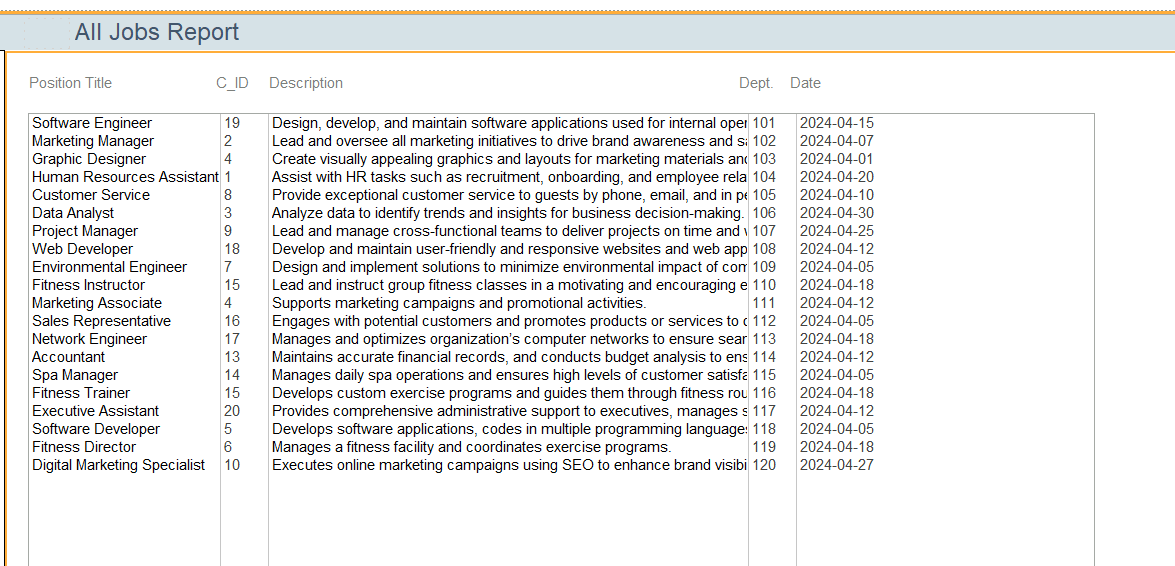


Here is the SQL command we used to get the results above:

*SELECT \* FROM Application;*

***Jobs Report***

This form will allow access to see jobs, with their title, company id, description, department, and date.



Here is the SQL command we used to get the results above:

*SELECT \* FROM Job;*

VII. **Conclusions (Ahamed)**

To approach the heightened demand for our HR company as it provides efficient management of various aspects of human resources, such as recruiting, paperwork, training, payroll processing, performance reviews, and compliance/legal support, we have formulated a comprehensive database management system. This system is meant to respond to the various hurdles related to the use of spreadsheets on Excel. To target this long standing goal, we aimed to provide a centralized and scalable solution.

This multifactorial plan includes the implementation of several modules for lead management, hiring, payroll inventory, performance review tracking, and system security under the umbrella of this database management system. In doing so, this allows us to trace the company’s leads, hiring leads, payroll inventory, performance review information, and system security on a practical and efficient scale. Moreover, the system assists data analysis in terms of customizable reports, allowing us to get some insight within the hiring approach, company leads, and payroll efficiencies.

Shifting into this database management framework, we expect to improve the overall operational efficiency and client satisfaction while positioning ourselves for long term prosperity and success within the competitive HR field. Looking at the framework’s scalability, flexibility, and robust security measures, it is evident that these aspects can support the company's extensions and evolving business requirements with time. Outside of this, the promise to provide training and support to clients will play a role in allowing them to fully leverage the system's resources.

In all, the database management system highlights a remarkable growth for the company's expedition towards modernizing HR management processes and delivering superior services to our clients.