Database Systems Project

HR & Company Management System

(Microsoft SQL Server Edition)

Prepared by :

Aya Abdelaziz **Basmalla Mabrouk**

Gehad Ahmed **Mariam Ashraf**

Supervised by: Eng.Marwa Elgazaly

Project Overview

The **Company Management & HR System** is an integrated relational database solution developed to streamline and centralize corporate human resources operations. This project is implemented using **Microsoft SQL Server (T-SQL)** and features a structured design that includes normalized tables, referential integrity constraints, and optimized stored procedures
The system is engineered to manage the entire employee journey—from initial **Recruitment** and **Employment History** tracking to daily **Attendance**, **Leave Requests**, and monthly **Payroll** processing. Additionally, it incorporates **Performance** evaluations and **Training** development modules to ensure professional growth and organizational efficiency.

Objectives

- **Centralization**: Securely store and centralize complete employee data in a single source of truth.
- **Automation**: Streamline and automate critical HR functions, specifically attendance tracking, leave management, and monthly payroll processing.
- **Reporting**: Provide HR managers and leadership with fast, accurate, and actionable reports.
- **Data Integrity & Security**: Ensure high data integrity, confidentiality, and controlled access through strict constraints and role-based permissions.
- **Operational Efficiency**: Reduce manual administrative overhead and improve overall HR operational efficiency.
- **Scalability**: Support future organizational growth, scalability, and seamless integration with external systems.

Scope

The system is divided into several integrated modules to cover the full spectrum of HR operations:

- **Employee Information Management**: Detailed tracking of personal, contact, and job-specific data.
- **Attendance & Leave Tracking**: Monitoring daily check-ins/outs and managing the workflow for leave requests and balances.
- **Payroll Management**: Automating salary calculations, including allowances and deductions.
- **Training & Development**: Managing employee enrollment, progress, and certification in corporate training programs.
- **Reporting Module**: Generating summaries for attendance, payroll, and employee master lists.
- **Role-Based Access Control (RBAC)**: Securing the system through specific user roles (Admin, HR, Manager, Employee) with defined permissions

Database Relationships

The following table defines the logical connections between entities, ensuring referential integrity through Foreign Keys (FK).

Primary Table	Related Table	Relationship Type	Description
Department	Employee	1:N (One-to-Many)	Each department can have multiple employees.
Position	Employee	1:N (One-to-Many)	Each position can be held by multiple employees.
Employee	EmployeePhone	1:N (One-to-Many)	An employee can have more than one contact number.
Employee	Attendance	1:N (One-to-Many)	An employee has multiple daily attendance records.

Primary Table	Related Table	Relationship Type	Description
Employee	LeaveRequest	1:N (One-to-Many)	An employee can submit multiple leave applications.
Employee	EmploymentHistory	1:N (One-to-Many)	Tracks all previous roles and departments for an employee.
Training	Employee_Training	1:N (One-to-Many)	Each training course can have many enrolled employees.
Employee	Payroll	1:N (One-to-Many)	An employee receives monthly salary records.
AllowanceType	PayrollAllowance	1:N	Defines different types of allowances (Housing, Transport, etc.).
DeductionType	PayrollDeduction	1:N	Defines different types of deductions (Tax, Insurance, Late, etc.).
Payroll	PayrollAllowance	1:N	Links a specific monthly payroll record to multiple allowances.
Payroll	PayrollDeduction	1:N	Links a specific monthly payroll record to multiple deductions.

Database Dictionary

Table: Department :

Column Name	Data Type	PK	FK	Description / Notes
DepartmentID	INT IDENTITY(1,1)	✓	✗	Auto-increment department identifier
DepartmentName	NVARCHAR(100)	✗	✗	Name of department (NOT NULL)
Location	NVARCHAR(100)	✗	✗	Department physical location
ManagerID	INT	✗	✓	FK → Employee (department manager)

Table: Position :

Column Name	Data Type	PK	FK	Description / Notes
EmployeeID	INT IDENTITY	✓	✗	Unique employee ID
FirstName	NVARCHAR(50)	✗	✗	Employee first name
LastName	NVARCHAR(50)	✗	✗	Employee last name
Gender	NVARCHAR(10)	✗	✗	Male / Female
DateOfBirth	DATE	✗	✗	Birth date
Email	NVARCHAR(100)	✗	✗	Unique email address
Address	NVARCHAR(255)	✗	✗	Employee address

Column Name	Data Type	PK	FK	Description / Notes
HireDate	DATE	✗	✗	Hiring date
Salary	DECIMAL(10,2)	✗	✗	Current salary
Status	NVARCHAR(20)	✗	✗	Active / Inactive
DepartmentID	INT	✗	✓	FK → Department
PositionID	INT	✗	✓	FK → Position

Table: Employee :

Column Name	Data Type	PK	FK	Description / Notes
EmployeeID	INT IDENTITY	✓	✗	Unique employee ID
FirstName	NVARCHAR(50)	✗	✗	Employee first name
LastName	NVARCHAR(50)	✗	✗	Employee last name
Gender	NVARCHAR(10)	✗	✗	Male / Female
DateOfBirth	DATE	✗	✗	Birth date
Email	NVARCHAR(100)	✗	✗	Unique email address
Address	NVARCHAR(255)	✗	✗	Employee address
HireDate	DATE	✗	✗	Hiring date
Salary	DECIMAL(10,2)	✗	✗	Current salary
Status	NVARCHAR(20)	✗	✗	Active / Inactive
DepartmentID	INT	✗	✓	FK → Department
PositionID	INT	✗	✓	FK → Position

Table: EmployeePhone :

Column Name	Data Type	PK	FK	Description / Notes
PhoneID	INT IDENTITY	✓	✗	Phone record identifier
EmployeeID	INT	✗	✓	FK → Employee
Phone	NVARCHAR(20)	✗	✗	Employee phone number

Table: Attendance :

Column Name	Data Type	PK	FK	Description / Notes
PhoneID	INT IDENTITY	✓	✗	Phone record identifier
EmployeeID	INT	✗	✓	FK → Employee
Phone	NVARCHAR(20)	✗	✗	Employee phone number

Note: WorkingHours is derived from CheckIn and CheckOut.

Table: LeaveRequest :

Column Name	Data Type	PK	FK	Description / Notes
LeaveID	INT IDENTITY	✓	✗	Leave request identifier
EmployeeID	INT	✗	✓	FK → Employee
LeaveType	NVARCHAR(50)	✗	✗	Annual / Sick / Emergency
StartDate	DATE	✗	✗	Leave start date
EndDate	DATE	✗	✗	Leave end date
Status	NVARCHAR(20)	✗	✗	Pending / Approved / Rejected
ApprovedBy	INT	✗	✓	FK → Employee (manager)

Note: Total leave days calculated using DATEDIFF

Tabel Payroll:

Column Name	Data Type	PK	FK	Description / Notes
PayrollID	INT IDENTITY	✓	✗	Payroll record ID
EmployeeID	INT	✗	✓	FK → Employee
Month	INT	✗	✗	Payroll month

Column Name	Data Type	PK	FK	Description / Notes
Year	INT	✗	✗	Payroll year
BasicSalary	DECIMAL(10,2)	✗	✗	Base salary
Allowances	DECIMAL(10,2)	✗	✗	Additional payments
Deductions	DECIMAL(10,2)	✗	✗	Salary deductions
PaymentDate	DATE	✗	✗	Payment date

Note: NetSalary = BasicSalary + Allowances − Deductions (Derived)

Table: Performance:

Column Name	Data Type	PK	FK	Description / Notes
PerformanceID	INT IDENTITY	✓	✗	Performance review ID
EmployeeID	INT	✗	✓	Evaluated employee
ReviewDate	DATE	✗	✗	Review date
Rating	INT	✗	✗	Performance score
Comments	NVARCHAR(500)	✗	✗	Manager feedback
ReviewedBy	INT	✗	✓	FK → Employee (reviewer)
ReviewPeriod	NVARCHAR(50)	✗	✗	Evaluation period

Table: Training :

Column Name	Data Type	PK	FK	Description / Notes
TrainingID	INT IDENTITY	✓	✗	Training program ID
TrainingName	NVARCHAR(150)	✗	✗	Program name
Description	NVARCHAR(500)	✗	✗	Training details

Column Name	Data Type	PK	FK	Description / Notes
StartDate	DATE	✗	✗	Start date
EndDate	DATE	✗	✗	End date
TrainerID	INT	✗	✓	FK → Employee (trainer)

Table: Employee_Training :

Column Name	Data Type	PK	FK	Description / Notes
EmployeeID	INT	✓	✓	FK → Employee
TrainingID	INT	✓	✓	FK → Training
CompletionStatus	NVARCHAR(50)	✗	✗	Completed / Pending
Certificate	NVARCHAR(100)	✗	✗	Certificate info
Score	DECIMAL(5,2)	✗	✗	Training score

Table: Recruitment

Column Name	Data Type	PK	FK	Description / Notes
CandidateID	INT IDENTITY	✓	✗	Candidate identifier
FirstName	NVARCHAR(50)	✗	✗	Candidate first name
LastName	NVARCHAR(50)	✗	✗	Candidate last name
Email	NVARCHAR(100)	✗	✗	Candidate email
Phone	NVARCHAR(20)	✗	✗	Contact number
AppliedPositionID	INT	✗	✓	FK → Position
ApplicationDate	DATE	✗	✗	Apply date
Status	NVARCHAR(20)	✗	✗	Pending / Accepted / Rejected

Table: UserAccount

Column Name	Data Type	PK	FK	Description / Notes
UserID	INT IDENTITY	✓	✗	System user ID

Column Name	Data Type	PK	FK	Description / Notes
Username	NVARCHAR(50)	✗	✗	Login username (UNIQUE)
Password	NVARCHAR(200)	✗	✗	Encrypted password
Role	NVARCHAR(50)	✗	✗	Admin / HR / Employee
EmployeeID	INT	✗	✓	FK → Employee (1-to-1 relation)

Stored Procedures

All data manipulation and business logic are performed through stored procedures to ensure data consistency, security, and modularity.

Employee & Career Management

Procedure Name	Operation	Description
`SP_AddEmployee`	**INSERT**	Registers new employees with full personal and professional details.
`SP_UpdateEmployee`	**UPDATE**	Updates profiles using `COALESCE` for partial data updates.
`SP_DeleteEmployee`	**DELETE**	Performs a safe delete by removing related phone and history records first.
`SP_AddEmploymentHistory`	**INSERT/UPDATE**	Manages internal transfers by closing the old role and opening a new one.

Attendance Management

These procedures automate the daily time-tracking process for employees.

Procedure	Operation	Description
`CheckIn`	**INSERT**	Logs the start of the workday. It captures the current date and time, setting the initial status to 'Present'.
`CheckOut`	**UPDATE**	Records the end of the workday for a specific employee by updating the `CheckOut` column for the current date.

Leave Management

This module handles the submission and approval workflow for employee time-off requests.

Procedure	Operation	Description
ApplyLeave	**INSERT**	Allows an employee to submit a request for leave (Annual, Sick, etc.) with a default status of 'Pending'.
ApproveLeave	**UPDATE**	Updates a specific leave request to 'Approved' and records the ID of the manager who authorized it.
RejectLeave	**UPDATE**	Updates a specific leave request to 'Rejected' and records the manager's ID for accountability.

Training & Development

Procedure Name	Operation	Description
`sp_AssignEmployeeTraining`	**INSERT**	Enrolls an employee in a course and records initial progress.
`sp_UpdateTrainingScore`	**UPDATE**	Updates performance scores for specific training modules.
`sp_GetTrainingEmployees`	**SELECT**	Retrieves a list of all staff members enrolled in a specific course.

Advanced Payroll Business Logic

Procedure Name	Operation	Description
`SP_CreatePayroll`	**INSERT**	Creates a monthly payroll record, fetches basic salary, and triggers late deduction calculations.
`SP_AddPayrollAllowance`	**INSERT**	Assigns a specific allowance type and amount to an employee's payroll.
`SP_AddPayrollDeduction`	**INSERT**	Assigns a specific deduction type and amount to an employee's payroll.
`SP_ApplyLateDeduction`	**LOGIC/INSERT**	Scans attendance records for 'Late' status and automatically applies a penalty (50 per day) to the payroll.
`SP_GetEmployeePayHistory`	**SELECT**	Retrieves the full payment history for a specific employee using their ID.

Advanced Functions & Reporting (Views)

Reports are generated through dynamic SQL Views for real-time data visualization and analysis.

Custom Functions :

Function Name	Return Type	Purpose
`FN_CalcAge`	**INT**	Automatically calculates the current age of an employee based on `DateOfBirth`.

Analytical Reporting Views :

View Name	Category	Description
`vw_EmployeeMasterList`	**General**	A complete directory of employees with departments and grades.
`vw_AttendanceSummary`	**Attendance**	Statistics on attendance behavior (Present, Late, Absent).
`vw_PayrollReport`	**Finance**	Calculates Net Salary (`Basic + Allowances - Deductions`).
`vw_LeaveBalance`	**Leaves**	Tracks remaining leave days based on a 21-day annual limit.
`vw_TrainingDetails`	**Training**	Comprehensive report on employee training history and certification.
`vw_TrainingSchedule`	**Schedule**	Displays upcoming courses and assigned trainers.
`vw_CompletedTrainings`	**Awards**	A list of employees who successfully earned certificates.

Advanced Financial Functions

Function Name	Return Type	Purpose
`FN_TotalAllowances`	**DECIMAL**	Sums up all allowances assigned to a specific `PayrollID`.
`FN_TotalDeductions`	**DECIMAL**	Sums up all deductions assigned to a specific `PayrollID`.
`FN_NetSalary`	**DECIMAL**	Calculates final pay: `BasicSalary + TotalAllowances - TotalDeductions`.

Financial Reports

View Name	Category	Description
`V_PayrollSummary`	**Finance**	A high-level view showing Employee Name, Basic Salary, Total Allowances/Deductions, and Net Salary.

Conclusion

The **HR & Company Management System** project successfully demonstrates the power of relational databases in streamlining corporate operations. By leveraging **Microsoft SQL Server**, we have built a centralized platform that not only stores data but also automates complex business logic through **Stored Procedures** and provides actionable insights via **Analytical Views**.

The implementation of data integrity constraints, such as unique keys and check constraints, ensures a high level of data quality. Furthermore, the modular design of the system—covering Payroll, Attendance, Leaves, and Training—provides a scalable foundation that can be expanded in the future to include advanced features like AI-driven performance forecasting or mobile integration.

This project has been a significant opportunity to apply database design principles and T-SQL programming to solve real-world organizational challenges.