

Database-001

1. Project Overview & Description

The **Database-001** project is a robust and scalable database system designed to manage and organize data for a company that offers a wide range of services and sells various items. Built using **Microsoft SQL Server**, the database is normalized to the **Third Normal Form (3NF)** to ensure data integrity, minimize redundancy, and optimize performance. The system supports multiple business processes, including employee management, department organization, service tracking, item inventory management, and customer interactions. Advanced features such as **computed columns**, **triggers**, **stored procedures**, **views**, and **non-clustered indices** enhance functionality and provide valuable insights into the data.

The database (Database-001) is structured to manage various aspects of an IT services company's operations, including:

1. Company and organizational structure management
2. Employee information and skills tracking
3. Customer relationship management
4. Service delivery across multiple IT domains:
 - Cloud Solutions
 - Security Systems
 - Network Management
 - Software Development
 - Hardware and Software Repair Services
5. Inventory and item management
6. Service delivery tracking and logging

The database is designed to support a company that provides comprehensive IT services while maintaining detailed records of employees, customers, services, and inventory items.

2. Scope

The **Database-001** project covers the following key business processes:

Included Processes:

1. Company Management:

- Tracks company details such as name, trade registration number, and contact information.
- Links companies to departments, employees, services, and items.

2. Employee Management:

- Manages employee details, including personal information, job position, salary, and skills.
- Associates employees with specific departments and companies.

3. Department Management:

- Tracks department details, including the number of employees and mission statements.
- Links departments to specific companies.

4. Service Management:

- Manages services provided by the company, including service details, duration, pricing, and status.
- Associates services with specific departments and companies.

5. Item Management:

- Tracks items sold by the company, including item details, quantity, and price.
- Links items to specific companies.

6. Customer Management:

- Manages customer details and tracks their interactions with the company, such as service requests and item purchases.
- Associates customers with specific companies.

7. Logging and Auditing:

- Logs service-related actions (e.g., insertions) for auditing purposes.
-

Excluded Processes:

1. Financial Transactions:

- The database does not handle financial transactions such as billing, invoicing, or payment processing.

2. Customer Support Tickets:

- The database does not manage detailed customer support tickets or issue tracking.

3. Advanced Analytics:

- While the database includes basic views for data summarization, it does not include advanced analytics or reporting features.
-

3. Data and Requirements Analysis

The database was designed based on the following business requirements:

1. Company Requirements:

- Each company can have multiple departments, employees, services, and items.
- Companies must have unique trade registration numbers.

2. Employee Requirements:

- Employees must belong to a specific department and company.
- Employees can possess multiple skills, which are tracked in a separate table.

3. Department Requirements:

- Departments must belong to a specific company.
- Departments have a mission statement and track the number of employees.

4. Service Requirements:

- Services must be associated with a specific department and company.
- Services have a start date, end date, price, and status.
- Services can be categorized into different types (e.g., cloud, security, repair, network, software development).

5. Item Requirements:

- Items must be associated with a specific company.
- Items have a name, quantity, and price.

6. Customer Requirements:

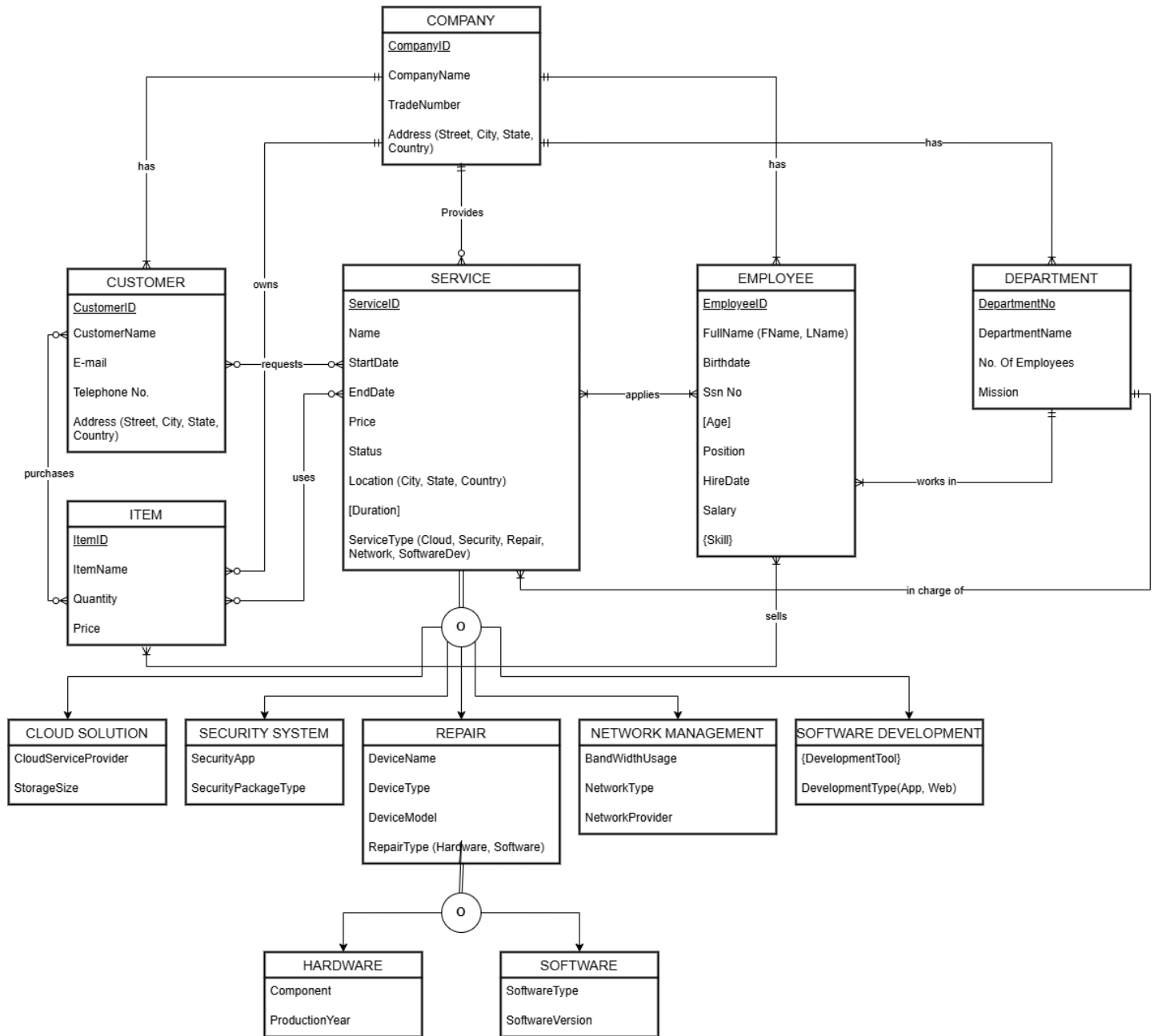
- Customers must be associated with a specific company.
- Customers can request services and purchase items.

7. Logging Requirements:

- All service-related actions (e.g., insertions) must be logged for auditing purposes.
-

4. Database Schema

The database consists of **21 tables**, **4 views**, **1 trigger**, **20 stored procedures**, and **4 non-clustered indices**.



Enhanced Entity Relationship Diagram

An Enhanced Entity-Relationship (EER) diagram for a database system related to a company. Here's a breakdown of the key components and relationships:

Entities:

1. COMPANY

- Attributes: CompanyID, CompanyName, TradeNumber, Address (Street, City, State, Country)
- Relationships: has CUSTOMER, provides SERVICE

2. CUSTOMER

- Attributes: CustomerID, CustomerName, E-mail, Telephone No., Address (Street, City, State, Country)
- Relationships: owns ITEM, requests SERVICE

3. SERVICE

- Attributes: ServiceID, Name, StartDate, EndDate, Price, Duration, ServiceType (Cloud, Security, Repair, Network, SoftwareDev)
- Relationships: provided by COMPANY, requested by CUSTOMER, uses ITEM

4. EMPLOYEE

- Attributes: EmployeeID, FullName (FName, LName), Birthdate, Sen No, Address (Street, City, State, Country), Position, HireDate, Salary, [Age], Status
- Relationships: works in DEPARTMENT, in charge of SERVICE

5. DEPARTMENT

- Attributes: DepartmentNo, DepartmentName, No.Of Employees, Mission, Location (City, State, Country)
- Relationships: has EMPLOYEE

6. ITEM

- Attributes: ItemID, ItemName, Quantity, Price
- Relationships: sold by COMPANY, purchased by CUSTOMER, used by SERVICE

Specialized Services:

1. CLOUD SOLUTION

- Attributes: CloudServiceProvider, StorageSize, BandWidthUsage

2. SECURITY SYSTEM

- Attributes: SecurityApp, SecurityPackageType

3. REPAIR

- Attributes: DeviceName, DeviceType, DeviceModel, RepairType (Hardware, Software)

4. NETWORK MANAGEMENT

- Attributes: NetworkType, NetworkProvider

5. SOFTWARE DEVELOPMENT

- Attributes: DevelopmentType (App, Web), DevelopmentTool

Subtypes of ITEM:

1. HARDWARE

- Attributes: Component, ProductionYear

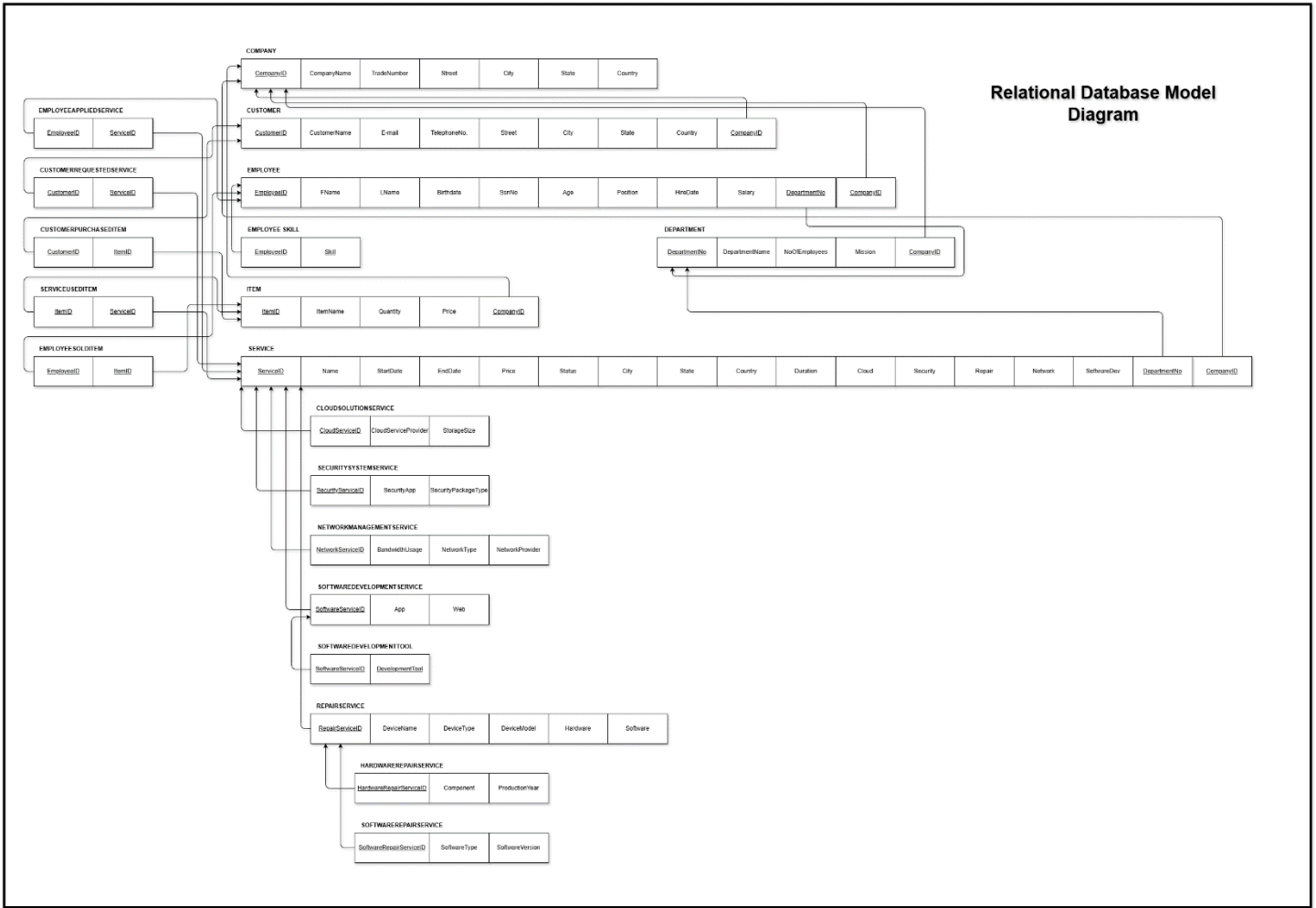
2. SOFTWARE

- Attributes: SoftwareType, SoftwareVersion

Relationships:

- **COMPANY** has **CUSTOMER** and provides **SERVICE**.
- **CUSTOMER** owns **ITEM** and requests **SERVICE**.
- **SERVICE** uses **ITEM**.
- **EMPLOYEE** works in **DEPARTMENT** and is in charge of **SERVICE**.
- **DEPARTMENT** has **EMPLOYEE**.

This EER diagram outlines the structure of a database system for managing company operations, including customer interactions, service provision, employee management, and item inventory. Each entity has specific attributes and relationships that define its role within the system.



Relational Database Model Diagram (3NF)

Below is a detailed description of each component:

Tables

1. Company

- **Fields/Columns:**
 - CompanyID (INT)
 - CompanyName (VARCHAR(50))
 - TradeNumber (INT)
 - Street (VARCHAR(50))
 - City (VARCHAR(50))
 - State (VARCHAR(50))
 - Country (VARCHAR(50))
 - **Definition:** Stores company information.
 - **Data Types:**
 - CompanyID: INT
 - CompanyName: VARCHAR(50)
 - TradeNumber: INT
 - Street: VARCHAR(50)
 - City: VARCHAR(50)
 - State: VARCHAR(50)
 - Country: VARCHAR(50)
 - **Indexes:** CompanyID (Primary Key).
 - **Constraints:** TradeNumber is UNIQUE.
-

2. Employee

- **Fields/Columns:**

- EmployeeID (INT)
- FirstName (VARCHAR(50))
- LastName (VARCHAR(50))
- Birthdate (DATE)
- SsnNo (VARCHAR(50))
- Age (Computed Column)
- Position (VARCHAR(50))
- HireDate (DATE)
- Salary (INT)
- DepartmentNo (INT)
- CompanyID (INT)

- **Definition:** Stores employee details, linked to a company and department.

- **Data Types:**

- EmployeeID: INT
- FirstName: VARCHAR(50)
- LastName: VARCHAR(50)
- Birthdate: DATE
- SsnNo: VARCHAR(50)
- Age: Computed Column
- Position: VARCHAR(50)
- HireDate: DATE
- Salary: INT
- DepartmentNo: INT
- CompanyID: INT

- **Indexes:** EmployeeID (Primary Key), CompanyID (Non-clustered Index).

- **Constraints:** (Salary > 0), SsnNo is UNIQUE, Foreign Key to Department and Company.

- **Computed Columns:** Age is computed based on Birthdate.
-

3. Department

- **Fields/Columns:**
 - DepartmentNo (INT)
 - DepartmentName (VARCHAR(50))
 - NoOfEmployees (INT)
 - Mission (VARCHAR(100))
 - CompanyID (INT)
 - **Definition:** Stores department details, linked to a company.
 - **Data Types:**
 - DepartmentNo: INT
 - DepartmentName: VARCHAR(50)
 - NoOfEmployees: INT
 - Mission: VARCHAR(100)
 - CompanyID: INT
 - **Indexes:** DepartmentNo (Primary Key), CompanyID (Non-clustered Index).
 - **Constraints:** Foreign Key to Company.
-

4. Service

- **Fields/Columns:**

- ServiceID (INT)
- Name (VARCHAR(50))
- StartDate (DATE)
- EndDate (DATE)
- Price (INT)
- Status (VARCHAR(50))
- City (VARCHAR(50))
- State (VARCHAR(50))
- Country (VARCHAR(50))
- Cloud (VARCHAR(50))
- Security (VARCHAR(50))
- Repair (VARCHAR(50))
- Network (VARCHAR(50))
- SoftwareDev (VARCHAR(50))
- Duration (Computed Column)
- CompanyID (INT)
- DepartmentNo (INT)

- **Definition:** Stores service details, linked to a company and department.

- **Data Types:**

- ServiceID: INT
- Name: VARCHAR(50)
- StartDate: DATE
- EndDate: DATE
- Price: INT
- Status: VARCHAR(50)
- City: VARCHAR(50)
- State: VARCHAR(50)
- Country: VARCHAR(50)

- Cloud: VARCHAR(50)
 - Security: VARCHAR(50)
 - Repair: VARCHAR(50)
 - Network: VARCHAR(50)
 - SoftwareDev: VARCHAR(50)
 - Duration: Computed Column
 - CompanyID: INT
 - DepartmentNo: INT
 - **Indexes:** ServiceID (Primary Key), CompanyID (Non-clustered Index).
 - **Constraints:** Foreign Key to Company and Department.
 - **Computed Columns:** Duration is computed as the difference between EndDate and StartDate.
-

5. Item

- **Fields/Columns:**
 - ItemID (INT)
 - ItemName (VARCHAR(50))
 - Quantity (INT)
 - Price (INT)
 - CompanyID (INT)
 - **Definition:** Stores item details, linked to a company.
 - **Data Types:**
 - ItemID: INT
 - ItemName: VARCHAR(50)
 - Quantity: INT
 - Price: INT
 - CompanyID: INT
 - **Indexes:** ItemID (Primary Key), CompanyID (Non-clustered Index).
 - **Constraints:** Foreign Key to Company.
-

6. Customer

- **Fields/Columns:**
 - CustomerID (INT)
 - CustomerName (VARCHAR(50))
 - Email (VARCHAR(50))
 - TelephoneNo (VARCHAR(50))
 - Street (VARCHAR(50))
 - City (VARCHAR(50))
 - State (VARCHAR(50))
 - Country (VARCHAR(50))
 - CompanyID (INT)
 - **Definition:** Stores customer details, linked to a company.
 - **Data Types:**
 - CustomerID: INT
 - CustomerName: VARCHAR(50)
 - Email: VARCHAR(50)
 - TelephoneNo: VARCHAR(50)
 - Street: VARCHAR(50)
 - City: VARCHAR(50)
 - State: VARCHAR(50)
 - Country: VARCHAR(50)
 - CompanyID: INT
 - **Indexes:** CustomerID (Primary Key).
 - **Constraints:** Foreign Key to Company.
-

7. EmployeeSkill

- **Fields/Columns:**
 - EmployeeID (INT)
 - Skill (VARCHAR(50))
 - **Definition:** Tracks skills of employees.
 - **Data Types:**
 - EmployeeID: INT
 - Skill: VARCHAR(50)
 - **Indexes:** Composite Primary Key (EmployeeID, Skill).
 - **Constraints:** Foreign Key to Employee.
-

8. EmployeeAppliedService

- **Fields/Columns:**
 - EmployeeID (INT)
 - ServiceID (INT)
 - **Definition:** Tracks which employees applied specific services.
 - **Data Types:**
 - EmployeeID: INT
 - ServiceID: INT
 - **Indexes:** Composite Primary Key (EmployeeID, ServiceID).
 - **Constraints:** Foreign Key to Employee and Service.
-

9. CustomerRequestedService

- **Fields/Columns:**
 - CustomerID (INT)
 - ServiceID (INT)
 - **Definition:** Tracks which customers requested specific services.
 - **Data Types:**
 - CustomerID: INT
 - ServiceID: INT
 - **Indexes:** Composite Primary Key (CustomerID, ServiceID).
 - **Constraints:** Foreign Key to Customer and Service.
-

10. CustomerPurchasedItem

- **Fields/Columns:**
 - CustomerID (INT)
 - ItemID (INT)
 - **Definition:** Tracks which customers purchased specific items.
 - **Data Types:**
 - CustomerID: INT
 - ItemID: INT
 - **Indexes:** Composite Primary Key (CustomerID, ItemID).
 - **Constraints:** Foreign Key to Customer and Item.
-

11. ServiceUsedItem

- **Fields/Columns:**
 - ServiceID (INT)
 - ItemID (INT)
 - **Definition:** Tracks which items were used in specific services.
 - **Data Types:**
 - ServiceID: INT
 - ItemID: INT
 - **Indexes:** Composite Primary Key (ServiceID, ItemID).
 - **Constraints:** Foreign Key to Service and Item.
-

12. EmployeeSoldItem

- **Fields/Columns:**
 - EmployeeID (INT)
 - ItemID (INT)
 - **Definition:** Tracks which employees sold specific items.
 - **Data Types:**
 - EmployeeID: INT
 - ItemID: INT
 - **Indexes:** Composite Primary Key (EmployeeID, ItemID).
 - **Constraints:** Foreign Key to Employee and Item.
-

13. ServiceLog

- **Fields/Columns:**
 - LogID (INT)
 - ServiceID (INT)
 - Action (VARCHAR(50))
 - LogDate (DATETIME)
 - **Definition:** Logs service-related actions.
 - **Data Types:**
 - LogID: INT (Identity)
 - ServiceID: INT
 - Action: VARCHAR(50)
 - LogDate: DATETIME (Default: GETDATE())
 - **Indexes:** LogID (Primary Key).
 - **Constraints:** Foreign Key to Service.
 - **Defaults:** LogDate defaults to the current date and time.
-

14. CloudSolutionService

- **Fields/Columns:**
 - CloudServiceID (INT)
 - CloudServiceProvider (VARCHAR(50))
 - StorageSize (VARCHAR(50))
 - **Definition:** Subtype of Service for cloud-related services.
 - **Data Types:**
 - CloudServiceID: INT
 - CloudServiceProvider: VARCHAR(50)
 - StorageSize: VARCHAR(50)
 - **Indexes:** CloudServiceID (Primary Key).
 - **Constraints:** Foreign Key to Service.
-

15. SecuritySystemService

- **Fields/Columns:**
 - SecurityServiceID (INT)
 - SecurityApp (VARCHAR(50))
 - SecurityPackageType (VARCHAR(50))
 - **Definition:** Subtype of Service for security-related services.
 - **Data Types:**
 - SecurityServiceID: INT
 - SecurityApp: VARCHAR(50)
 - SecurityPackageType: VARCHAR(50)
 - **Indexes:** SecurityServiceID (Primary Key).
 - **Constraints:** Foreign Key to Service.
-

16. NetworkManagementService

- **Fields/Columns:**
 - NetworkServiceID (INT)
 - BandwidthUsage (VARCHAR(50))
 - NetworkType (VARCHAR(50))
 - NetworkProvider (VARCHAR(50))
 - **Definition:** Subtype of Service for network-related services.
 - **Data Types:**
 - NetworkServiceID: INT
 - BandwidthUsage: VARCHAR(50)
 - NetworkType: VARCHAR(50)
 - NetworkProvider: VARCHAR(50)
 - **Indexes:** NetworkServiceID (Primary Key).
 - **Constraints:** Foreign Key to Service.
-

17. SoftwareDevelopmentService

- **Fields/Columns:**
 - SoftwareServiceID (INT)
 - App (VARCHAR(50))
 - Web (VARCHAR(50))
 - **Definition:** Subtype of Service for software development-related services.
 - **Data Types:**
 - SoftwareServiceID: INT
 - App: VARCHAR(50)
 - Web: VARCHAR(50)
 - **Indexes:** SoftwareServiceID (Primary Key).
 - **Constraints:** Foreign Key to Service.
-

18. SoftwareDevelopmentTool

- **Fields/Columns:**
 - SoftwareServiceID (INT)
 - DevelopmentTool (VARCHAR(50))
 - **Definition:** Tracks development tools used in software development services.
 - **Data Types:**
 - SoftwareServiceID: INT
 - DevelopmentTool: VARCHAR(50)
 - **Indexes:** Composite Primary Key (SoftwareServiceID, DevelopmentTool).
 - **Constraints:** Foreign Key to SoftwareDevelopmentService.
-

19. RepairService

- **Fields/Columns:**
 - RepairServiceID (INT)
 - DeviceName (VARCHAR(50))
 - DeviceType (VARCHAR(50))
 - DeviceModel (VARCHAR(50))
 - Hardware (VARCHAR(50))
 - Software (VARCHAR(50))
 - **Definition:** Subtype of Service for repair-related services.
 - **Data Types:**
 - RepairServiceID: INT
 - DeviceName: VARCHAR(50)
 - DeviceType: VARCHAR(50)
 - DeviceModel: VARCHAR(50)
 - Hardware: VARCHAR(50)
 - Software: VARCHAR(50)
 - **Indexes:** RepairServiceID (Primary Key).
 - **Constraints:** Foreign Key to Service.
-

20. HardwareRepairService

- **Fields/Columns:**
 - HardwareRepairServiceID (INT)
 - Component (VARCHAR(50))
 - ProductionYear (INT)
 - **Definition:** Subtype of RepairService for hardware repair services.
 - **Data Types:**
 - HardwareRepairServiceID: INT
 - Component: VARCHAR(50)
 - ProductionYear: INT
 - **Indexes:** HardwareRepairServiceID (Primary Key).
 - **Constraints:** Foreign Key to RepairService.
-

21. SoftwareRepairService

- **Fields/Columns:**
 - SoftwareRepairServiceID (INT)
 - SoftwareType (VARCHAR(50))
 - SoftwareVersion (VARCHAR(50))
 - **Definition:** Subtype of RepairService for software repair services.
 - **Data Types:**
 - SoftwareRepairServiceID: INT
 - SoftwareType: VARCHAR(50)
 - SoftwareVersion: VARCHAR(50)
 - **Indexes:** SoftwareRepairServiceID (Primary Key).
 - **Constraints:** Foreign Key to RepairService.
-

Views

1. CompanyOverview

- **Definition:** Summarizes key company data, including the total number of employees, departments, services, and items for each company.

2. DepartmentEmployeeOverview

- **Definition:** Provides an overview of employee data within each department, including the total number of employees, average salary, and highest and lowest salary.

3. ServiceStatusOverview

- **Definition:** Provides an overview of the status of services offered by each company, including the total number of services, total service duration, and average service duration.

4. DepartmentItemOverview

- **Definition:** Summarizes item data for each department under the same company, including the total number of items, average item price, and most and least expensive items.
-

Triggers

1. trg_AfterInsert_Service

- **Definition:** This trigger fires after an INSERT operation on the **Service** table and logs the action in the **ServiceLog** table.
 - **When/How it works:** After a new service is inserted, the trigger automatically logs the action in the **ServiceLog** table, recording the ServiceID, action type (INSERT), and the current date and time.
-

Stored Procedures

1. InsertCompany

- **Definition:** Inserts a new company into the **Company** table.

2. GetCompany

- **Definition:** Retrieves company information based on the provided CompanyID.

3. UpdateCompany

- **Definition:** Updates company information based on the provided CompanyID.

4. DeleteCompany

- **Definition:** Deletes a company from the **Company** table based on the provided CompanyID.

5. InsertCustomer

- **Definition:** Inserts a new customer into the **Customer** table.

6. GetCustomer

- **Definition:** Retrieves customer information based on the provided CustomerID.

7. UpdateCustomer

- **Definition:** Updates customer information based on the provided CustomerID.

8. DeleteCustomer

- **Definition:** Deletes a customer from the **Customer** table based on the provided CustomerID.

9. InsertDepartment

- **Definition:** Inserts a new department into the **Department** table.

10. GetDepartment

- **Definition:** Retrieves department information based on the provided DepartmentNo.

11. UpdateDepartment

- **Definition:** Updates department information based on the provided DepartmentNo.

12. DeleteDepartment

- **Definition:** Deletes a department from the **Department** table based on the provided DepartmentNo.

13. InsertItem

- **Definition:** Inserts a new item into the **Item** table.

14. GetItem

- **Definition:** Retrieves item information based on the provided ItemID.

15. UpdateItem

- **Definition:** Updates item information based on the provided ItemID.

16. DeleteItem

- **Definition:** Deletes an item from the **Item** table based on the provided ItemID.

17. InsertService

- **Definition:** Inserts a new service into the **Service** table.

18. GetService

- **Definition:** Retrieves service information based on the provided ServiceID.

19. UpdateService

- **Definition:** Updates service information based on the provided ServiceID.

20. DeleteService

- **Definition:** Deletes a service from the **Service** table based on the provided ServiceID.

5. Conclusion

The **Database-001** project successfully implements a normalized database schema to manage company, employee, department, service, item, and customer data. The database includes advanced features such as **views**, **triggers**, **stored procedures**, and **non-clustered indices** to support various business processes and ensure data integrity. The project meets the requirements outlined in the project guidelines and is ready for further development and integration with business applications. The system provides a solid foundation for efficient data management and business process automation.
