

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:

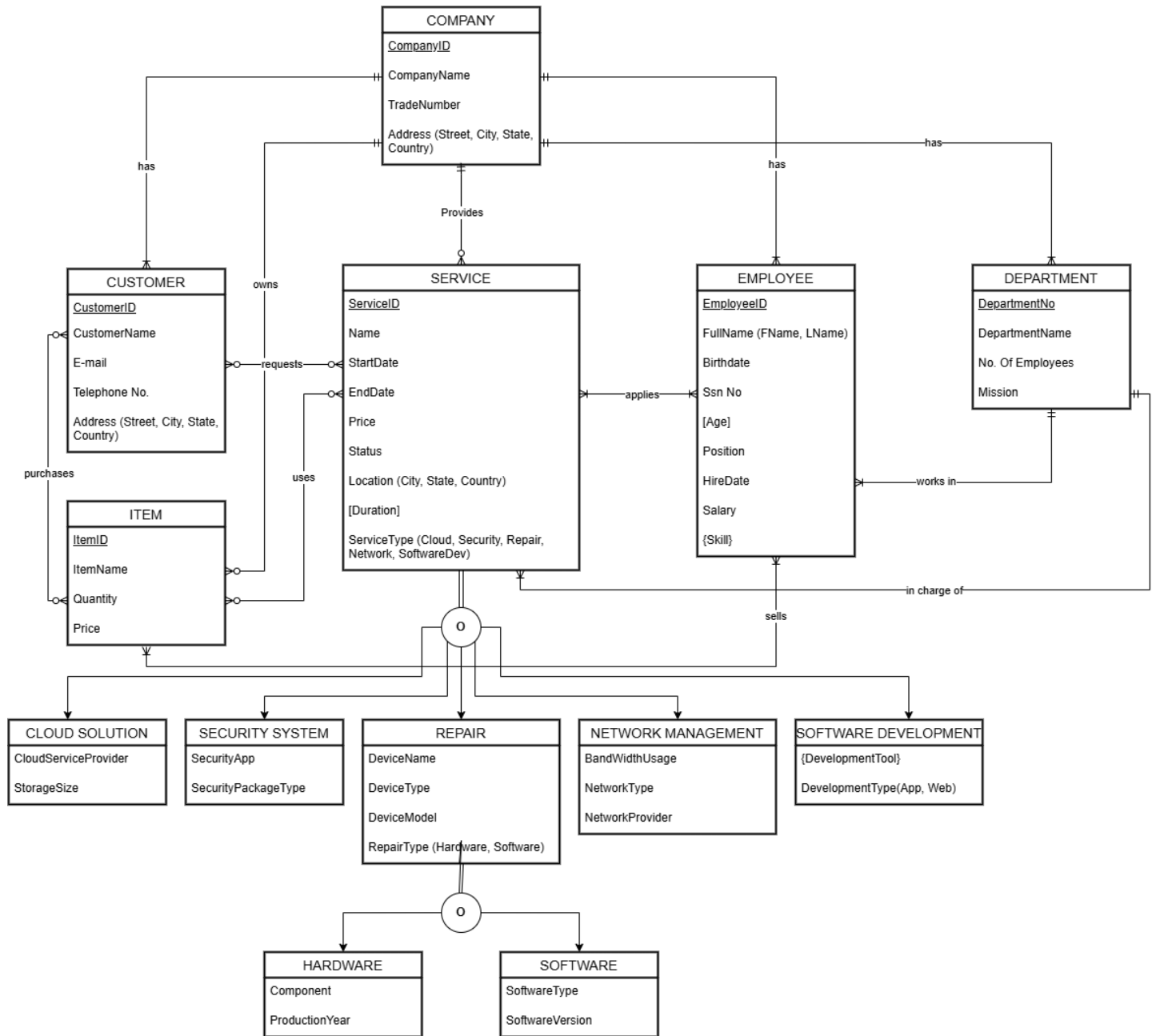
- Customers must be associated with a specific company.
- Customers can request services and purchase items.

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

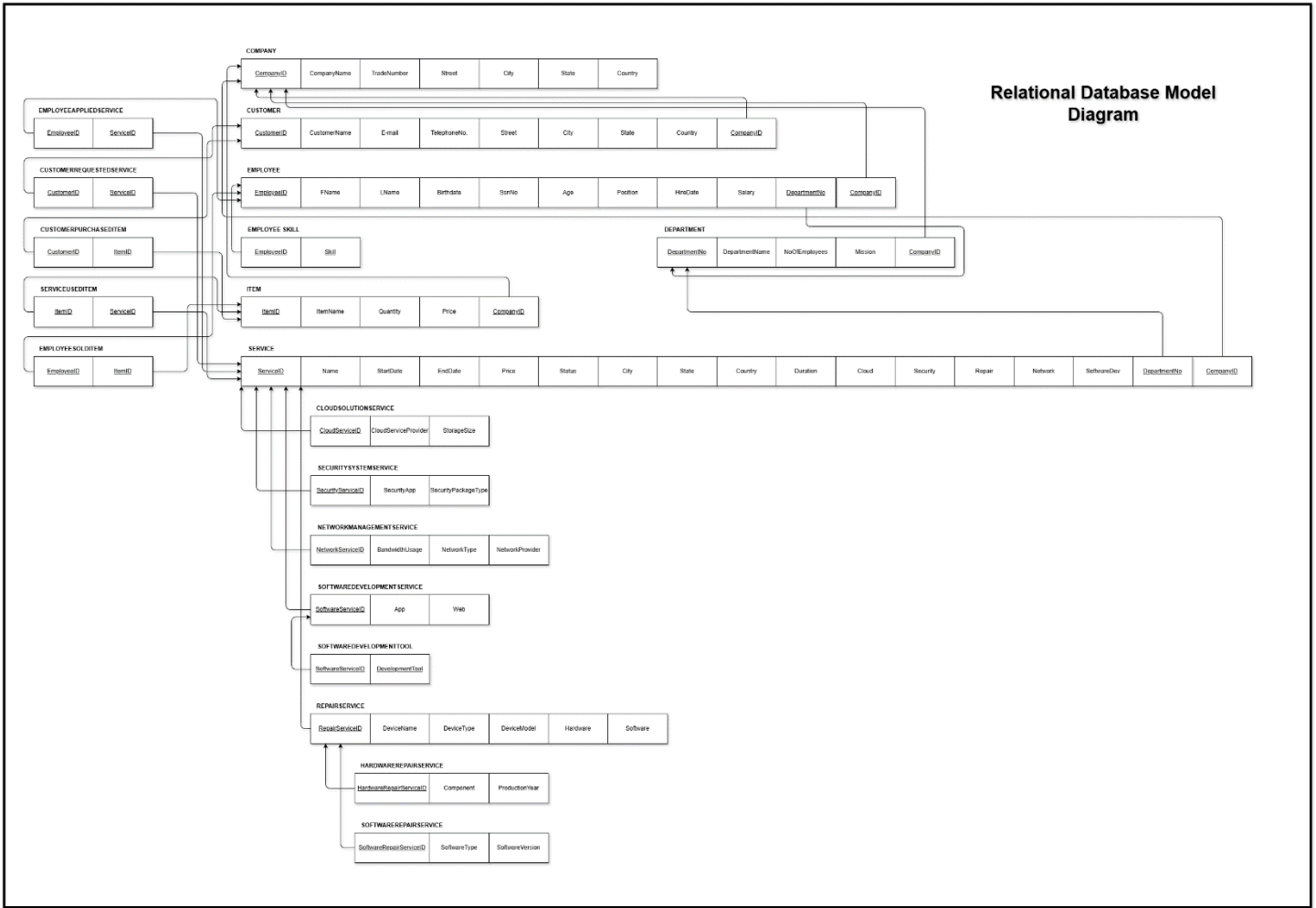
2. SOFTWARE

- 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.

```
USE [Database-001]
GO

/***** Object: View [dbo].[CompanyOverview]    Script Date: 1/3/2025 6:16:36 PM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [dbo].[CompanyOverview] AS
SELECT
    c.CompanyID,
    c.CompanyName,
    COUNT(DISTINCT e.EmployeeID) AS [TotalEmployees],
    COUNT(DISTINCT d.DepartmentNo) AS [TotalDepartments],
    COUNT(DISTINCT s.ServiceID) AS [TotalServices],
    COUNT(DISTINCT i.ItemID) AS [TotalItems]
FROM
    Company c
LEFT JOIN Employee e ON c.CompanyID = e.CompanyID
LEFT JOIN Department d ON c.CompanyID = d.CompanyID
LEFT JOIN Service s ON c.CompanyID = s.CompanyID
LEFT JOIN Item i ON c.CompanyID = i.CompanyID
GROUP BY
    c.CompanyID, c.CompanyName;
GO

SELECT * FROM CompanyOverview;
```

100 %

Results Messages

	CompanyID	CompanyName	TotalEmployees	TotalDepartments	TotalServices	TotalItems
1	1	Atasehir Bilisim LTD	50	15	210	30

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.

```
USE [Database-001]
GO

/***** Object: View [dbo].[DepartmentEmployeeOverview]    Script Date: 1/3/2025 6:27:53 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [dbo].[DepartmentEmployeeOverview] AS
SELECT
    d.DepartmentNo,
    d.DepartmentName,
    d.Mission,
    COUNT(e.EmployeeID) AS [TotalEmployees],
    AVG(e.Salary) AS [AverageSalary],
    MAX(e.Salary) AS [HighestSalary],
    MIN(e.Salary) AS [LowestSalary]
FROM
    Department d
JOIN Employee e ON d.DepartmentNo = e.DepartmentNo
GROUP BY
    d.DepartmentNo, d.DepartmentName, d.Mission;
GO

SELECT * FROM DepartmentEmployeeOverview;
```

100 %

Results Messages

	DepartmentNo	DepartmentName	Mission	TotalEmployees	AverageSalary	HighestSalary	LowestSalary
1	1	Administration	To oversee daily operations and coordination	6	59666	70000	48000
2	2	Cloud Services	To provide reliable cloud-based solutions and supp...	4	66500	75000	62000
3	3	Network Management	To maintain and optimize network infrastructure	4	62500	67000	60000
4	4	Software Repair	To troubleshoot and fix software-related issues	1	48000	48000	48000
5	5	Hardware Repair	To repair and maintain hardware devices	2	61000	70000	52000
6	6	Security Systems	To ensure robust security and protection against th...	4	62500	68000	58000
7	7	Software Development	To design and build innovative software solutions	13	57076	63000	50000
8	8	Data Analytics	To analyze data for insights and decision-making	5	60400	70000	52000
9	9	Customer Support	To assist customers with their technical and service...	4	57500	75000	48000
10	10	Testing and Quality Assurance	To test and ensure the quality of services and prod...	2	52500	55000	50000
11	11	Research and Development	To innovate and develop next-generation technolo...	2	66000	67000	65000
12	12	Human Resources	To manage and support the company's workforce	2	49000	50000	48000
13	13	Finance and Accounting	To handle financial transactions and reporting	1	45000	45000	45000

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.

```
USE [Database-001]
GO

/***** Object: View [dbo].[ServiceStatusOverview]    Script Date: 1/3/2025 6:35:52 PM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [dbo].[ServiceStatusOverview] AS
SELECT
    s.CompanyID,
    c.CompanyName,
    s.Status AS ServiceStatus,
    COUNT(s.ServiceID) AS [TotalServices],
    SUM(s.Duration) AS [TotalServiceDuration],
    AVG(s.Duration) AS [AverageServiceDuration]
FROM
    Service s
JOIN Company c ON s.CompanyID = c.CompanyID
GROUP BY
    s.CompanyID, c.CompanyName, s.Status;
GO

SELECT * FROM ServiceStatusOverview;
```

100 %

Results Messages

	CompanyID	CompanyName	ServiceStatus	TotalServices	TotalServiceDuration	AverageServiceDuration
1	1	Atasehir Bilisim LTD	Active	8	104	13
2	1	Atasehir Bilisim LTD	Completed	202	1709	8

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.

```
USE [Database-001]
GO

/***** Object: View [dbo].[DepartmentItemOverview]    Script Date: 1/3/2025 6:29:29 PM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [dbo].[DepartmentItemOverview] AS
SELECT
    -- Department details
    d.DepartmentNo,
    d.DepartmentName,
    d.Mission,

    -- Company details
    c.CompanyID,
    c.CompanyName,

    -- Count the total number of items associated with the company's departments
    COUNT(i.ItemID) AS [TotalItems],

    -- Calculate the average price of items in the company's departments
    ISNULL(AVG(i.Price), 0) AS [AverageItemPrice],

    -- Find the most expensive item in the company's departments
    ISNULL(MAX(i.Price), 0) AS [MostExpensiveItemPrice],

    -- Find the least expensive item in the company's departments
    ISNULL(MIN(i.Price), 0) AS [LeastExpensiveItemPrice],

    -- Identify the name of the most expensive item
    (SELECT TOP 1 i1.ItemName
     FROM Item i1
     JOIN Department d1 ON d1.CompanyID = i1.CompanyID
     WHERE d1.DepartmentNo = d.DepartmentNo
     ORDER BY i1.Price DESC) AS [MostExpensiveItemName],

    -- Identify the name of the least expensive item
    (SELECT TOP 1 i2.ItemName
     FROM Item i2
     JOIN Department d2 ON d2.CompanyID = i2.CompanyID
     WHERE d2.DepartmentNo = d.DepartmentNo
     ORDER BY i2.Price ASC) AS [LeastExpensiveItemName]

FROM
    -- Base table is the Department table
    Department d

    -- Join the Company table to link departments to their respective companies
```

```

FROM
    -- Base table is the Department table
    Department d

    -- Join the Company table to link departments to their respective companies
    JOIN Company c ON d.CompanyID = c.CompanyID

    -- Join the Item table to link items to their respective companies
    LEFT JOIN Item i ON c.CompanyID = i.CompanyID

    -- Group by department and company details to aggregate item data
    GROUP BY
        d.DepartmentNo,
        d.DepartmentName,
        d.Mission,
        c.CompanyID,
        c.CompanyName;
GO

```

```

SELECT * FROM DepartmentEmployeeOverview;

```

-- Views

```

SELECT * FROM CompanyOverview;

SELECT * FROM DepartmentEmployeeOverview;

SELECT * FROM DepartmentItemOverview;

SELECT * FROM ServiceStatusOverview;

```

100 %

Results Messages

	DepartmentNo	DepartmentName	Mission	CompanyID	CompanyName	TotalItems	AverageItemPrice	MostExpensiveItemPrice	LeastExpensiveItemPrice	MostExpensiveItemName	LeastExpensiveItemName
1	1	Administration	To oversee daily operations and coordination	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
2	2	Cloud Services	To provide reliable cloud-based solutions and supp...	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
3	3	Network Management	To maintain and optimize network infrastructure	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
4	4	Software Repair	To troubleshoot and fix software-related issues	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
5	5	Hardware Repair	To repair and maintain hardware devices	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
6	6	Security Systems	To ensure robust security and protection against th...	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
7	7	Software Development	To design and build innovative software solutions	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
8	8	Data Analytics	To analyze data for insights and decision-making	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
9	9	Customer Support	To assist customers with their technical and service...	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
10	10	Testing and Quality Assurance	To test and ensure the quality of services and prod...	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
11	11	Research and Development	To innovate and develop next-generation technolo...	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
12	12	Human Resources	To manage and support the company's workforce	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
13	13	Finance and Accounting	To handle financial transactions and reporting	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
14	14	Marketing and Sales	To promote and sell the company's products and s...	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive
15	15	Legal and Compliance	To ensure legal compliance and protect the compa...	1	Atasehir Bilisim LTD	30	350	1500	10	Refrigerator	USB Flash Drive

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.

```
USE [Database-001]
GO

/***** Object: Trigger [dbo].[trg_AfterInsert_Service]    Script Date: 1/3/2025 11:23:49 PM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-----
-- Trigger: trg_AfterInsert_Service
-----
-- This trigger fires after an INSERT operation on the Service table.
-- It logs the action in the ServiceLog table.
-----
CREATE TRIGGER [dbo].[trg_AfterInsert_Service]
ON [dbo].[Service]
AFTER INSERT
AS
BEGIN
    -- Insert a log entry for each new service added
    INSERT INTO ServiceLog (ServiceID, Action, LogDate)
    SELECT ServiceID, 'INSERT', GETDATE()
    FROM INSERTED;
END;
GO

ALTER TABLE [dbo].[Service] ENABLE TRIGGER [trg_AfterInsert_Service]
GO
```

```
-- Triggers
SELECT * FROM Service;
INSERT INTO Service(ServiceID, Name, StartDate, EndDate, Price, Status, City, State, Country, Cloud, Security, Repair, Network, SoftwareDev, CompanyID, DepartmentNo)
VALUES(1, 'Special Servie', '2025-01-01', '2026-01-01', 100, 'Active', 'Istanbul', 'Esenyurt', 'Turkey', null, null, null, null, 'SoftwareDev', 1, 2);

SELECT * FROM ServiceLog;
EXEC DeleteService @ServiceID = 1;
SELECT * FROM Service;
SELECT * FROM ServiceLog;
```

100 %

Results Messages

	ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo
1	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
2	2002	Cloud Backup Services	2023-02-01	2023-02-13	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	12	1	2
3	2003	Cloud Storage	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
4	2004	Cloud Application Hosting	2023-04-01	2023-04-13	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	12	1	2
5	2005	Cloud Security Solutions	2023-05-01	2023-05-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
6	2006	Private Cloud Solutions	2023-06-01	2023-06-14	4500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
7	2007	Cloud Platform Development	2023-07-01	2023-07-14	6000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
8	2008	Disaster Recovery Cloud	2023-08-01	2023-08-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
9	2009	Cloud Data Analytics	2023-09-01	2023-09-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
10	2010	Cloud DevOps Services	2023-10-01	2023-10-14	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
11	2011	Cloud Migration Solutions	2023-11-01	2023-11-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
12	2012	Cloud API Integration	2023-12-01	2023-12-14	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
13	2013	Hybrid Cloud Solutions	2023-01-15	2023-01-29	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	14	1	2
14	2014	Cloud Infrastructure Management	2023-02-15	2023-02-28	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
15	2015	Multi-Cloud Solutions	2023-03-15	2023-03-28	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
16	2016	Cloud Computing Solutions	2023-04-15	2023-04-28	6000	Active	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
17	2017	Cloud Database Solutions	2023-05-15	2023-05-28	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
18	2018	Cloud Storage for Enterprises	2023-06-15	2023-06-28	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
19	2019	Cloud-Based Business Solutions	2023-07-15	2023-07-28	5000	Active	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
20	2020	Cloud Disaster Recovery	2023-08-15	2023-08-28	6000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2

Query executed successfully.

localhost\MSSQLSERVI

```
-- Triggers
SELECT * FROM Service;
INSERT INTO Service(ServiceID, Name, StartDate, EndDate, Price, Status, City, State, Country, Cloud, Security, Repair, Network, SoftwareDev, CompanyID, DepartmentNo)
VALUES(1, 'Special Servie', '2025-01-01', '2026-01-01', 100, 'Active', 'Istanbul', 'Esenyurt', 'Turkey', null, null, null, null, 'SoftwareDev', 1, 2);

SELECT * FROM ServiceLog;
EXEC DeleteService @ServiceID = 1;
SELECT * FROM Service;
SELECT * FROM ServiceLog;
```

100 %

Messages

(1 row affected)

(1 row affected)

Completion time: 2025-01-02T21:48:01.7469204+03:00

```
-- Triggers
SELECT * FROM Service;
INSERT INTO Service(ServiceID, Name, StartDate, EndDate, Price, Status, City, State, Country, Cloud, Security, Repair, Network, SoftwareDev, CompanyID, DepartmentNo)
VALUES(1, 'Special Servie', '2025-01-01', '2026-01-01', 100, 'Active', 'Istanbul', 'Esenyurt', 'Turkey', null, null, null, null, 'SoftwareDev', 1, 2);

SELECT * FROM ServiceLog;
SELECT * FROM Service;
EXEC DeleteService @ServiceID = 1;
SELECT * FROM Service;
SELECT * FROM ServiceLog;
```

100 %

Results Messages

LogID	ServiceID	Action	LogDate
1	3	INSERT	2025-01-03 21:48:01.723

	ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo
1	1	Special Servie	2025-01-01	2026-01-01	100	Active	Istanbul	Esenyurt	Turkey	NULL	NULL	NULL	NULL	SoftwareDev	365	1	2
2	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
3	2002	Cloud Backup Services	2023-02-01	2023-02-13	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	12	1	2
4	2003	Cloud Storage	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
5	2004	Cloud Application Hosting	2023-04-01	2023-04-13	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	12	1	2
6	2005	Cloud Security Solutions	2023-05-01	2023-05-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
7	2006	Private Cloud Solutions	2023-06-01	2023-06-14	4500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
8	2007	Cloud Platform Development	2023-07-01	2023-07-14	6000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
9	2008	Disaster Recovery Cloud	2023-08-01	2023-08-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
10	2009	Cloud Data Analytics	2023-09-01	2023-09-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
11	2010	Cloud DevOps Services	2023-10-01	2023-10-14	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
12	2011	Cloud Migration Solutions	2023-11-01	2023-11-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
13	2012	Cloud API Integration	2023-12-01	2023-12-14	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
14	2013	Hybrid Cloud Solutions	2023-01-15	2023-01-29	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	14	1	2
15	2014	Cloud Infrastructure Mana...	2023-02-15	2023-02-28	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
16	2015	Multi-Cloud Solutions	2023-03-15	2023-03-28	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2

Query executed successfully.

localhost\MSSQLSERVI

```
-- Triggers
SELECT * FROM Service;
INSERT INTO Service(ServiceID, Name, StartDate, EndDate, Price, Status, City, State, Country, Cloud, Security, Repair, Network, SoftwareDev, CompanyID, DepartmentNo)
VALUES(1, 'Special Servie', '2025-01-01', '2026-01-01', 100, 'Active', 'Istanbul', 'Esenyurt', 'Turkey', null, null, null, null, 'SoftwareDev', 1, 2);

SELECT * FROM ServiceLog;
SELECT * FROM Service;
EXEC DeleteService @ServiceID = 1;
SELECT * FROM Service;
SELECT * FROM ServiceLog;
```

100 %

Results Messages

	ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo
1	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
2	2002	Cloud Backup Services	2023-02-01	2023-02-13	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	12	1	2
3	2003	Cloud Storage	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
4	2004	Cloud Application Hosting	2023-04-01	2023-04-13	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	12	1	2
5	2005	Cloud Security Solutions	2023-05-01	2023-05-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
6	2006	Private Cloud Solutions	2023-06-01	2023-06-14	4500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
7	2007	Cloud Platform Development	2023-07-01	2023-07-14	6000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
8	2008	Disaster Recovery Cloud	2023-08-01	2023-08-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
9	2009	Cloud Data Analytics	2023-09-01	2023-09-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
10	2010	Cloud DevOps Services	2023-10-01	2023-10-14	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
11	2011	Cloud Migration Solutions	2023-11-01	2023-11-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
12	2012	Cloud API Integration	2023-12-01	2023-12-14	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
13	2013	Hybrid Cloud Solutions	2023-01-15	2023-01-29	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	14	1	2

LogID	ServiceID	Action	LogDate
-------	-----------	--------	---------

Stored Procedures

1. InsertCompany

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[InsertCompany]    Script Date: 1/3/2025 11:29:16 PM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-----
-- #####
-- PROCEDURES FOR COMPANY TABLE CRUD OPERATIONS
-- These procedures perform Create, Read, Update, and Delete (CRUD) operations
-- on the 'Company' table with appropriate error handling and transactional support.
-- #####
-----

-- InsertCompany: Inserts a new company into the Company table.
-- Parameters:
-- @CompanyID (INT): Unique identifier for the company.
-- @CompanyName (VARCHAR(50)): Name of the company.
-- @TradeNumber (INT): Trade number of the company.
-- @Street (VARCHAR(50)): Street address of the company.
-- @City (VARCHAR(50)): City of the company.
-- @State (VARCHAR(50)): State of the company.
-- @Country (VARCHAR(50)): Country of the company.
-----

CREATE PROCEDURE [dbo].[InsertCompany]
    @CompanyID INT,
    @CompanyName VARCHAR(50),
    @TradeNumber INT,
    @Street VARCHAR(50),
    @City VARCHAR(50),
    @State VARCHAR(50),
    @Country VARCHAR(50)
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the CompanyID already exists in the Company table.
        IF EXISTS (SELECT 1 FROM Company WHERE CompanyID = @CompanyID)
        BEGIN
            THROW 50001, 'CompanyID already exists. Insert operation aborted.', 1;
        END;

        -- Perform the insert operation.
        INSERT INTO Company (CompanyID, CompanyName, TradeNumber, Street, City, State, Country)
        VALUES (@CompanyID, @CompanyName, @TradeNumber, @Street, @City, @State, @Country);

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
```

75 %

Connected. (1/1)


```
-- Company Procedures
SELECT * FROM Company;
-- Insert
EXEC InsertCompany
    @CompanyID = 201,
    @CompanyName = 'Tech Innovators Inc.',
    @TradeNumber = 12345,
    @Street = '789 Innovation Blvd',
    @City = 'Metropolis',
    @State = 'CA',
    @Country = 'USA';

SELECT * FROM Company;

-- Delete
EXEC DeleteCompany
    @CompanyID = 201;
```

100 %

Results Messages

	CompanyID	CompanyName	TradeNumber	Street	City	State	Country
1	1	Atasehir Bilisim LTD	991282728	MOR SÜMBÜL SK.	ISTANBUL	ATASEHIR	Türkiye

```
-- Procedures
-- Company Procedures
SELECT * FROM Company;
-- Insert
EXEC InsertCompany
    @CompanyID = 201,
    @CompanyName = 'Tech Innovators Inc.',
    @TradeNumber = 12345,
    @Street = '789 Innovation Blvd',
    @City = 'Metropolis',
    @State = 'CA',
    @Country = 'USA';

SELECT * FROM Company;

-- Delete
EXEC DeleteCompany
    @CompanyID = 201;
```

100 %

Results Messages

	CompanyID	CompanyName	TradeNumber	Street	City	State	Country
1	1	Atasehir Bilisim LTD	991282728	MOR SÜMBÜL SK.	ISTANBUL	ATASEHIR	Türkiye
2	201	Tech Innovators Inc.	12345	789 Innovation Blvd	Metropolis	CA	USA

2. GetCompany

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[GetCompany]    Script Date: 1/4/2025 12:07:32 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

--=====
-- GetCompany: Retrieves company information.
-- Parameters:
--   @CompanyID (INT, optional): The ID of the company to retrieve.
--   If NULL, retrieves all companies.
--=====

CREATE PROCEDURE [dbo].[GetCompany]
    @CompanyID INT = NULL
AS
BEGIN
    BEGIN TRY
        -- If no CompanyID is provided, select all records.
        IF @CompanyID IS NULL
        BEGIN
            SELECT * FROM Company;
        END
        ELSE
        BEGIN
            -- Check if the specified CompanyID exists.
            IF NOT EXISTS (SELECT 1 FROM Company WHERE CompanyID = @CompanyID)
            BEGIN
                THROW 50002, 'CompanyID not found. Read operation aborted.', 1;
            END;

            -- Retrieve the company record matching the provided CompanyID.
            SELECT * FROM Company WHERE CompanyID = @CompanyID;
        END;
    END TRY
    BEGIN CATCH
        -- Handle any errors that occur during the read operation.
        THROW;
    END CATCH;
END;
GO
```

```
-- get
SELECT * FROM Company;

EXEC GetCompany
    @CompanyID = 1;

SELECT * FROM Company;
```

100 %

Results Messages

	CompanyID	CompanyName	TradeNumber	Street	City	State	Country
1	1	Atasehir Bilisim LTD	991282728	MOR SÜMBÜL SK.	ISTANBUL	ATASEHIR	Türkiye

3. UpdateCompany

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[UpdateCompany]    Script Date: 1/4/2025 12:27:12 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-----
-- UpdateCompany: Updates company information.
-- Parameters:
--   @CompanyID (INT): The ID of the company to update.
--   @CompanyName, @TradeNumber, @Street, @City, @State, @Country (optional):
--     Fields to update. If NULL, the field remains unchanged.
-----

CREATE PROCEDURE [dbo].[UpdateCompany]
    @CompanyID INT,
    @CompanyName VARCHAR(50) = NULL,
    @TradeNumber INT = NULL,
    @Street VARCHAR(50) = NULL,
    @City VARCHAR(50) = NULL,
    @State VARCHAR(50) = NULL,
    @Country VARCHAR(50) = NULL
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the specified CompanyID exists.
        IF NOT EXISTS (SELECT 1 FROM Company WHERE CompanyID = @CompanyID)
        BEGIN
            THROW 50003, 'CompanyID not found. Update operation aborted.', 1;
        END;

        -- Perform the update operation, using ISNULL to retain current values for NULL parameters.
        UPDATE Company
        SET
            CompanyName = ISNULL(@CompanyName, CompanyName),
            TradeNumber = ISNULL(@TradeNumber, TradeNumber),
            Street = ISNULL(@Street, Street),
            City = ISNULL(@City, City),
            State = ISNULL(@State, State),
            Country = ISNULL(@Country, Country)
        WHERE CompanyID = @CompanyID;

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
```

```
-- Update
SELECT * FROM Company;
EXEC InsertCompany
    @CompanyID = 201,
    @CompanyName = 'Tech Innovators Inc.',
    @TradeNumber = 12345,
    @Street = '789 Innovation Blvd',
    @City = 'Metropolis',
    @State = 'CA',
    @Country = 'USA';
SELECT * FROM Company;

-- Full Update
EXEC UpdateCompany
    @CompanyID = 201,
    @CompanyName = 'Tech Innovators Limited',
    @TradeNumber = 67890,
    @Street = '456 Research Park Drive',
    @City = 'Innovation City',
    @State = 'NY',
    @Country = 'USA';
```

100 %

Results Messages

	CompanyID	CompanyName	TradeNumber	Street	City	State	Country
1	1	Atasehir Bilisim LTD	991282728	MOR SÜMBÜL SK.	ISTANBUL	ATASEHIR	Türkiye
2	201	Tech Innovators Inc.	12345	789 Innovation Blvd	Metropolis	CA	USA

```
-- Full Update
EXEC UpdateCompany
    @CompanyID = 201,
    @CompanyName = 'Tech Innovators Limited',
    @TradeNumber = 67890,
    @Street = '456 Research Park Drive',
    @City = 'Innovation City',
    @State = 'NY',
    @Country = 'USA';
SELECT * FROM Company;
```

100 %

Results Messages

	CompanyID	CompanyName	TradeNumber	Street	City	State	Country
1	1	Atasehir Bilisim LTD	991282728	MOR SÜMBÜL SK.	ISTANBUL	ATASEHIR	Türkiye
2	201	Tech Innovators Inc.	12345	789 Innovation Blvd	Metropolis	CA	USA

```
SELECT * FROM Company;
```

```
SELECT * FROM Company;
```

```
-- Partial update:
```

```
EXEC UpdateCompany
```

```
    @CompanyID = 201,
```

```
    @CompanyName = 'Tech Updates Co.',
```

```
    @City = 'Silicon Valley';
```

```
SELECT * FROM Company;
```

100 %

Results Messages

	CompanyID	CompanyName	TradeNumber	Street	City	State	Country
1	1	Atasehir Bilisim LTD	991282728	MOR SÜMBÜL SK.	ISTANBUL	ATASEHIR	Türkiye
2	201	Tech Innovators Inc.	12345	789 Innovation Blvd	Metropolis	CA	USA

```
-- Partial update:
```

```
EXEC UpdateCompany
```

```
    @CompanyID = 201,
```

```
    @CompanyName = 'Tech Updates Co.',
```

```
    @City = 'Silicon Valley';
```

```
SELECT * FROM Company;
```

100 %

Results Messages

	CompanyID	CompanyName	TradeNumber	Street	City	State	Country
1	1	Atasehir Bilisim LTD	991282728	MOR SÜMBÜL SK.	ISTANBUL	ATASEHIR	Türkiye
2	201	Tech Updates Co.	12345	789 Innovation Blvd	Silicon Valley	CA	USA

4. DeleteCompany

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[DeleteCompany]    Script Date: 1/4/2025 12:06:43 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[DeleteCompany]
    @CompanyID INT
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the specified CompanyID exists.
        IF NOT EXISTS (SELECT 1 FROM Company WHERE CompanyID = @CompanyID)
        BEGIN
            THROW 50004, 'CompanyID not found. Delete operation aborted.', 1;
        END;

        -- Perform the delete operation.
        DELETE FROM Company WHERE CompanyID = @CompanyID;

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
```

```
-- Procedures
-- Company Procedures
SELECT * FROM Company;
-- Insert
EXEC InsertCompany
    @CompanyID = 201,
    @CompanyName = 'Tech Innovators Inc.',
    @TradeNumber = 12345,
    @Street = '789 Innovation Blvd',
    @City = 'Metropolis',
    @State = 'CA',
    @Country = 'USA';

SELECT * FROM Company;

-- Delete
EXEC DeleteCompany
    @CompanyID = 201;
```

100 %

Results Messages

	CompanyID	CompanyName	TradeNumber	Street	City	State	Country
1	1	Atasehir Bilisim LTD	991282728	MOR SÜMBÜL SK.	ISTANBUL	ATASEHIR	Türkiye
2	201	Tech Innovators Inc.	12345	789 Innovation Blvd	Metropolis	CA	USA

```
-- Delete
EXEC DeleteCompany
    @CompanyID = 201;

SELECT * FROM Company;
```

100 %

Results Messages

	CompanyID	CompanyName	TradeNumber	Street	City	State	Country
1	1	Atasehir Bilisim LTD	991282728	MOR SÜMBÜL SK.	ISTANBUL	ATASEHIR	Türkiye

5. InsertCustomer

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[InsertCustomer]    Script Date: 1/4/2025 12:35:52 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[InsertCustomer]
    @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
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the CustomerID already exists in the Customer table.
        IF EXISTS (SELECT 1 FROM Customer WHERE CustomerID = @CustomerID)
        BEGIN
            THROW 50005, 'CustomerID already exists. Insert operation aborted.', 1;
        END;

        -- Perform the insert operation.
        INSERT INTO Customer (CustomerID, CustomerName, Email, TelephoneNo, Street, City, State, Country, CompanyID)
        VALUES (@CustomerID, @CustomerName, @Email, @TelephoneNo, @Street, @City, @State, @Country, @CompanyID);

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
```

100 %

Connected. (1/1) | localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 0 rows


```
-- Customer Procedures
SELECT * FROM Customer;
--
EXEC InsertCustomer
    @CustomerID = 10001,
    @CustomerName = 'John Doe',
    @Email = 'johndoe@example.com',
    @TelephoneNo = '123-456-7890',
    @Street = '123 Elm Street',
    @City = 'Springfield',
    @State = 'IL',
    @Country = 'USA',
    @CompanyID = 1;

SELECT * FROM Customer;
```

100 %

Results Messages

	CustomerID	CustomerName	Email	TelephoneNo	Street	City	State	Country	CompanyI
482	482	Jose Johnston	bonniethomas@yahoo.com	001-252-045-65...	219 Amy Well	Margaretport	Louisiana	Poland	1
483	483	Heidi Ferguson	alexanderholmes@yahoo....	001-765-213-28...	7112 Timothy Inlet	Donmouth	Rhode Is...	Maldi...	1
484	484	Gloria Mokinn...	walkermelissa@watts.com	216.110.7603	89071 Williams Hollow	Port Colinville	Rhode Is...	Costa...	1
485	485	Christina Lam...	thomasjulie@rush-bailey.c...	938.911.9083x3...	87592 Nunez Coves	Port Michael	Missouri	Jamai...	1
486	486	Darren Rodri...	owensapril@yahoo.com	001-548-015-71...	6317 White Spring Suit...	North Douglasfurt	Vermont	Namibia	1
487	487	Natalie Stafford	coxjames@larson.com	674.994.6579	8427 David Overpass ...	Lake Brianland	New York	Philp...	1
488	488	Christopher J...	kevinhanson@gmail.com	+1-877-347-784...	67567 Johnson Fall Sui...	West Jennifer	New York	Serbia	1
489	489	Michael Keller	khart@hotmail.com	6479695450	025 Pacheco Prairie	Port Patriciamouth	Missouri	Bhutan	1
490	490	William Jenkins	johnwong@miller.com	208-507-6986x4...	9659 Lindsey Courts A...	Garrettport	Mississippi	Jordan	1
491	491	Mark Dickson	karlwilliams@morales.com	001-919-541-98...	468 Gonzalez Locks Su...	North Courtney	North Da...	Tokelau	1
492	492	Jonathan Gu...	sharon89@yahoo.com	001-083-572-3246	1364 Lucas Glens Apt. ...	East Anthony	Maryland	Kyrgy...	1
493	493	Dr. Javier Bul...	josephjohnson@hopkins-t...	001-700-159-41...	818 Crawford Course	New David	Massach...	Gree...	1
494	494	Nancy Shaffer	grobinsont@lewis.info	(490)785-1586x...	540 Mckenzie Shore A...	Parkerstad	Utah	British...	1
495	495	Cynthia Ramos	qpennington@davis-carso...	327-560-4177x6...	481 Fisher Stream	Shawntown	Wisconsin	Austria	1
496	496	Catherine Ra...	moralescourtney@johnsto...	296.577.2429x5...	66930 Sandra Rue Apt....	Jacquelineburgh	Arkansas	Iran	1
497	497	Lindsay Hern...	staceywilson@hotmail.com	677.133.3447x6...	3626 Anthony Greens	New Shannon	Kansas	Heard...	1
498	498	Maria Miller	monicamccann@yahoo.com	+1-366-941-180...	48964 Lambert Curve	New Harryburgh	North Ca...	Lithu...	1
499	499	Sherry Ander...	hthomas@yahoo.com	2745644948	2572 Brown Drive Apt. ...	Greenmouth	Nevada	Brazil	1
500	500	Tammy Willia...	uvasquez@gmail.com	513-422-8499	089 Robert Club	Port Tyler	Connecti...	Tuvalu	1

Query executed successfully.

localhost\MSSQLSERVER01 (16... PIRATE-960\Pirate-960 ... Database-001 00:00:00 500 rows

```
-- Customer Procedures
SELECT * FROM Customer;
--
EXEC InsertCustomer
    @CustomerID = 10001,
    @CustomerName = 'John Doe',
    @Email = 'johndoe@example.com',
    @TelephoneNo = '123-456-7890',
    @Street = '123 Elm Street',
    @City = 'Springfield',
    @State = 'IL',
    @Country = 'USA',
    @CompanyID = 1;

SELECT * FROM Customer;
```

100 %

Results Messages

	CustomerID	CustomerName	Email	TelephoneNo	Street	City	State	Country	CompanyID
483	483	Heidi Ferguson	alexanderholmes@yahoo...	001-765-213-28...	7112 Timothy Inlet	Donmouth	Rhode Is...	Maldi...	1
484	484	Gloria Mckinn...	walkermelissa@watts.com	216.110.7603	89071 Williams Hollow	Port Colinvile	Rhode Is...	Costa...	1
485	485	Christina Lam...	thomasjulia@rush-bailey.c...	938.911.9083x3...	87592 Nunez Coves	Port Michael	Missouri	Jamai...	1
486	486	Darren Rodri...	owensapril@yahoo.com	001-548-015-71...	6317 White Spring Suit...	North Douglasfurt	Vermont	Namibia	1
487	487	Natalie Stafford	coojames@larson.com	674.994.6579	8427 David Overpass ...	Lake Brianland	New York	Philip...	1
488	488	Christopher J...	kevinhanson@gmail.com	+1-877-347-784...	67567 Johnson Fall Sui...	West Jennifer	New York	Serbia	1
489	489	Michael Keller	khart@hotmail.com	6479695450	025 Pacheco Prairie	Port Patriciamouth	Missouri	Bhutan	1
490	490	William Jenkins	johnwong@miller.com	208-507-6986x4...	9659 Lindsey Courts A...	Garrettport	Mississippi	Jordan	1
491	491	Mark Dickson	karlwilliams@morales.com	001-919-541-98...	468 Gonzalez Looks Su...	North Courtney	North Da...	Tokelau	1
492	492	Jonathan Gu...	sharon89@yahoo.com	001-083-572-3246	1364 Lucas Glens Apt. ...	East Anthony	Maryland	Kyrgy...	1
493	493	Dr. Javier Bul...	josephjohnson@hopkins-t...	001-700-159-41...	818 Crawford Course	New David	Massach...	Gree...	1
494	494	Nancy Shaffer	grobinsont@lewis.info	(490)785-1586x...	540 McKenzie Shore A...	Parkerstad	Utah	British...	1
495	495	Cynthia Ramos	qpennington@davis-carso...	327-560-4177x6...	481 Fisher Stream	Shawntown	Wisconsin	Austria	1
496	496	Catherine Ra...	moralescourtney@johnsto...	296.577.2429x5...	66930 Sandra Rue Apt....	Jacquelineburgh	Arkansas	Iran	1
497	497	Lindsay Hern...	staceywilson@hotmail.com	677.133.3447x6...	3626 Anthony Greens	New Shannon	Kansas	Heard...	1
498	498	Maria Miller	monicamccann@yahoo.com	+1-366-941-180...	48964 Lambert Curve	New Harryburgh	North Ca...	Lithu...	1
499	499	Sherry Ander...	hthomas@yahoo.com	2745644948	2572 Brown Drive Apt. ...	Greenmouth	Nevada	Brazil	1
500	500	Tammy Willia...	uvasquez@gmail.com	513-422-8499	089 Robert Club	Port Tyler	Connecti...	Tuvalu	1
501	10001	John Doe	johndoe@example.com	123-456-7890	123 Elm Street	Springfield	IL	USA	1

Query executed successfully.

localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 501 rows

6. GetCustomer

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[GetCustomer]    Script Date: 1/4/2025 12:43:56 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[GetCustomer]
    @CustomerID INT = NULL
AS
BEGIN
    BEGIN TRY
        -- If no CustomerID is provided, select all records.
        IF @CustomerID IS NULL
        BEGIN
            SELECT * FROM Customer;
        END
        ELSE
        BEGIN
            -- Check if the specified CustomerID exists.
            IF NOT EXISTS (SELECT 1 FROM Customer WHERE CustomerID = @CustomerID)
            BEGIN
                THROW 50006, 'CustomerID not found. Read operation aborted.', 1;
            END;

            -- Retrieve the customer record matching the provided CustomerID.
            SELECT * FROM Customer WHERE CustomerID = @CustomerID;
        END;
    END TRY
    BEGIN CATCH
        -- Handle any errors that occur during the read operation.
        THROW;
    END CATCH;
END;
GO
```

```
-- Get
SELECT * FROM Customer;

-- Example usage -- full table:
EXEC GetCustomer;

-- Example usage with parameter - specific record:
EXEC GetCustomer
    @CustomerID = 101;
```

100 %

Results Messages

	CustomerID	CustomerName	Email	TelephoneNo	Street	City	State	Country	CompanyID
1	1	Ahmet Yılmaz	ahmet.yilmaz@gmail.com	+90 532 123 4567	Atatürk Bulvarı No:45 D:3	Çankaya	Ankara	Türkiye	1
2	2	Ayşe Kaya	ayse.kaya@hotmail.com	+90 532 987 6543	Konak Mahallesi	İnönü Caddesi No:12	İzmir	Türkiye	1
3	3	Fatma Demir	fatma.demir@yahoo.com	+90 531 456 7890	Liman Mahallesi	Lara Caddesi No:8	Antalya	Türkiye	1
4	4	Hüseyin Sahin	huseyin.sahin@outlook.com	+90 534 123 9876	Heykel Meydanı	Osmangazi	Bursa	Türkiye	1
5	5	Murat Çelik	murat.celik@gmail.com	+90 535 654 3210	Bagdat Caddesi No:56	Kadıköy	İstanbul	Türkiye	1
6	6	Zeynep Arslan	zeynep.arslan@icloud.com	+90 530 112 2334	Kalkınma Mahallesi	Farabi Sokak No:7	Trabzon	Türkiye	1
7	7	Burak Özkan	burak.ozkan@gmail.com	+90 536 998 7654	Turgut Özal Bulvarı No:89	Seyhan	Adana	Türkiye	1
8	8	Emine Kara	emine.kara@yahoo.com	+90 532 445 6677	Mevlana Caddesi No:34	Selçuklu	Konya	Türkiye	1
9	9	Sevgi Güneş	sevgi.gunes@gmail.com	+90 531 765 4321	Akdeniz Mahallesi	Atatürk Caddesi No:22	Mersin	Türkiye	1
10	10	Ali Polat	ali.polat@outlook.com	+90 530 567 1234	Kale Mahallesi	Cumhuriyet Caddesi No:3	Samsun	Türkiye	1
11	11	Ece Doğan	ece.dogan@gmail.com	+90 534 765 8910	Sehitkamil Mahallesi	Gazi Caddesi No:18	Gaziantep	Türkiye	1
12	12	Selin Aksoy	selin.aksoy@yahoo.com	+90 535 678 9102	Antakya Mahallesi	Asi Sokak No:5	Hatay	Türkiye	1
13	13	Tolga Eren	tolga.eren@hotmail.com	+90 532 987 0123	Deliktas Mahallesi	Çamlık Sokak No:19	Denizli	Türkiye	1
14	14	Esra Yıldız	esra.yildiz@gmail.com	+90 531 345 6789	Balıkesir Mahallesi	Yıldırım Sokak No:44	Balıkesir	Türkiye	1
15	15	Volkan Akin	volkan.akin@icloud.com	+90 536 123 4567	Yakutiye Mahallesi	Erzurum Caddesi No:11	Erzurum	Türkiye	1
16	16	Nur Gök	nur.gok@gmail.com	+90 530 876 5432	Tepebaşı Mahallesi	İstiklal Caddesi No:9	Eskişehir	Türkiye	1
17	17	Halil Arslan	halil.arslan@yahoo.com	+90 534 432 1098	Malatya Mahallesi	Şehir Caddesi No:28	Malatya	Türkiye	1
18	18	Gizem Kara	gizem.kara@outlook.com	+90 535 789 0123	Melikgazi Mahallesi	Cumhuriyet Sokak No:12	Kayseri	Türkiye	1
19	19	Cem Özdemir	cem.ozdemir@hotmail.com	+90 532 654 9871	Yenişehir Mahallesi	Çiftlik Sokak No:76	Sivas	Türkiye	1

Query executed successfully. | localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 500 rows

```
-- Example usage with parameter - specific record:
EXEC GetCustomer
    @CustomerID = 101;
```

100 %

Results Messages

	CustomerID	CustomerName	Email	TelephoneNo	Street	City	State	Country	CompanyID
1	101	Mehmet Çakır	mehmet.cakir@icloud.com	+90 532 334 4455	Eskişehir	Yunus Emre Caddesi No:11	Eskişehir	Türkiye	1

7. UpdateCustomer

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[UpdateCustomer]    Script Date: 1/4/2025 12:49:00 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[UpdateCustomer]
    @CustomerID INT,
    @CustomerName VARCHAR(50) = NULL,
    @Email VARCHAR(50) = NULL,
    @TelephoneNo VARCHAR(50) = NULL,
    @Street VARCHAR(50) = NULL,
    @City VARCHAR(50) = NULL,
    @State VARCHAR(50) = NULL,
    @Country VARCHAR(50) = NULL,
    @CompanyID INT = NULL
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the specified CustomerID exists.
        IF NOT EXISTS (SELECT 1 FROM Customer WHERE CustomerID = @CustomerID)
        BEGIN
            THROW 50007, 'CustomerID not found. Update operation aborted.', 1;
        END;

        -- Perform the update operation, using ISNULL to retain current values for NULL parameters.
        UPDATE Customer
        SET
            CustomerName = ISNULL(@CustomerName, CustomerName),
            Email = ISNULL(@Email, Email),
            TelephoneNo = ISNULL(@TelephoneNo, TelephoneNo),
            Street = ISNULL(@Street, Street),
            City = ISNULL(@City, City),
            State = ISNULL(@State, State),
            Country = ISNULL(@Country, Country),
            CompanyID = ISNULL(@CompanyID, CompanyID)
        WHERE CustomerID = @CustomerID;

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
    END CATCH
END
```

```

-- Commit the transaction if no errors occurred.
COMMIT TRANSACTION;
END TRY
BEGIN CATCH
-- Rollback the transaction in case of an error.
ROLLBACK TRANSACTION;
-- Re-throw the error for higher-level handling.
THROW;
END CATCH;
END;
GO

```

```

@CompanyID = 1;
SELECT * FROM Customer;
EXEC UpdateCustomer
@CustomerID = 10001,
@CustomerName = 'John A. Doe',
@email = 'john.adam.doe@example.com',
@TelephoneNo = '987-654-3210',

```

100 %

Results Messages

	CustomerID	CustomerName	Email	TelephoneNo	Street	City	State	Country	CompanyI
483	483	Heidi Ferguson	alexanderholmes@yahoo....	001-765-213-28...	7112 Timothy Inlet	Donmouth	Rhode Is...	Maldi...	1
484	484	Gloria Mckinn...	walkemelissa@watts.com	216.110.7603	89071 Williams Hollow	Port Colinville	Rhode Is...	Costa...	1
485	485	Christina Lam...	thomasjulia@rush-bailey.c...	938.911.9083x3...	87592 Nunez Coves	Port Michael	Missouri	Jamai...	1
486	486	Darren Rodri...	owensapril@yahoo.com	001-548-015-71...	6317 White Spring Suit...	North Douglasfurt	Vermont	Namibia	1
487	487	Natalie Stafford	coxjames@larson.com	674.994.6579	8427 David Overpass ...	Lake Brianland	New York	Philip...	1
488	488	Christopher J...	kevinhanson@gmail.com	+1-877-347-784...	67567 Johnson Fall Sui...	West Jennifer	New York	Serbia	1
489	489	Michael Keller	khart@hotmail.com	6479695450	025 Pacheco Prairie	Port Patriciamouth	Missouri	Bhutan	1
490	490	William Jenkins	johnwong@miller.com	208-507-6986x4...	9659 Lindsey Courts A...	Garrettport	Mississippi	Jordan	1
491	491	Mark Dickson	karlwilliams@morales.com	001-919-541-98...	468 Gonzalez Looks Su...	North Courtney	North Da...	Tokelau	1
492	492	Jonathan Gu...	sharon89@yahoo.com	001-083-572-3246	1364 Lucas Glens Apt. ...	East Anthony	Maryland	Kyrgy...	1
493	493	Dr. Javier Bul...	josephjohnson@hopkins-t...	001-700-159-41...	818 Crawford Course	New David	Massach...	Gree...	1
494	494	Nancy Shaffer	grobinson@lewis.info	(490)785-1586x...	540 Mckenzie Shore A...	Parkerstad	Utah	British...	1
495	495	Cynthia Ramos	qpennington@davis-carso...	327-560-4177x6...	481 Fisher Stream	Shawntown	Wisconsin	Austria	1
496	496	Catherine Ra...	moralescourtney@johnsto...	296.577.2429x5...	66930 Sandra Rue Apt....	Jacquelineburgh	Arkansas	Iran	1
497	497	Lindsay Hern...	staceywilson@hotmail.com	677.133.3447x6...	3626 Anthony Greens	New Shannon	Kansas	Heard...	1
498	498	Maria Miller	monicamocann@yahoo.com	+1-366-941-180...	48964 Lambert Curve	New Harryburgh	North Ca...	Lithu...	1
499	499	Sherry Ander...	hthomas@yahoo.com	2745644948	2572 Brown Drive Apt. ...	Greenmouth	Nevada	Brazil	1
500	500	Tammy Willia...	uvasquez@gmail.com	513-422-8499	089 Robert Club	Port Tyler	Connecti...	Tuvalu	1
501	10001	John Doe	johndoe@example.com	123-456-7890	123 Elm Street	Springfield	IL	USA	1

```

EXEC UpdateCustomer
    @CustomerID = 10001,
    @CustomerName = 'John A. Doe',
    @Email = 'john.adam.doe@example.com',
    @TelephoneNo = '987-654-3210',
    @Street = '456 Oak Street',
    @City = 'Shelbyville',
    @State = 'MO',
    @Country = 'USA',
    @CompanyID = 1;
SELECT * FROM Customer;

```

100 %

Results Messages

	CustomerID	CustomerName	Email	TelephoneNo	Street	City	State	Country	CompanyID
483	483	Heidi Ferguson	alexanderholmes@yahoo...	001-765-213-28...	7112 Timothy Inlet	Donmouth	Rhode Is...	Maldi...	1
484	484	Gloria McKinn...	walkermelissa@watts.com	216.110.7603	89071 Williams Hollow	Port Colinvile	Rhode Is...	Costa...	1
485	485	Christina Lam...	thomasjulia@rush-bailey.c...	938.911.9083x3...	87592 Nunez Coves	Port Michael	Missouri	Jamai...	1
486	486	Darren Rodri...	owensapril@yahoo.com	001-548-015-71...	6317 White Spring Suit...	North Douglasfurt	Vermont	Namibia	1
487	487	Natalie Stafford	coxjames@larson.com	674.994.6579	8427 David Overpass ...	Lake Brianland	New York	Philip...	1
488	488	Christopher J...	kevinhanson@gmail.com	+1-877-347-784...	67567 Johnson Fall Sui...	West Jennifer	New York	Serbia	1
489	489	Michael Keller	khart@hotmail.com	6479695450	025 Pacheco Prairie	Port Patriciamouth	Missouri	Bhutan	1
490	490	William Jenkins	johnwong@miller.com	208-507-6986x4...	9659 Lindsey Courts A...	Garrettport	Mississippi	Jordan	1
491	491	Mark Dickson	karlwilliams@morales.com	001-919-541-98...	468 Gonzalez Locks Su...	North Courtney	North Da...	Tokelau	1
492	492	Jonathan Gu...	sharon89@yahoo.com	001-083-572-3246	1364 Lucas Glens Apt. ...	East Anthony	Maryland	Kyrgy...	1
493	493	Dr. Javier Bul...	josephjohnson@hopkins-t...	001-700-159-41...	818 Crawford Course	New David	Massach...	Gree...	1
494	494	Nancy Shaffer	grobinsan@lewis.info	(490)785-1586x...	540 Mckenzie Shore A...	Parkerstad	Utah	British...	1
495	495	Cynthia Ramos	qpennington@davis-carso...	327-560-4177x6...	481 Fisher Stream	Shawntown	Wisconsin	Austria	1
496	496	Catherine Ra...	moralescourtney@johnsto...	296.577.2429x5...	66930 Sandra Rue Apt...	Jacquelineburgh	Arkansas	Iran	1
497	497	Lindsay Hern...	staceywilson@hotmail.com	677.133.3447x6...	3626 Anthony Greens	New Shannon	Kansas	Heard...	1
498	498	Maria Miller	monicamocann@yahoo.com	+1-366-941-180...	48964 Lambert Curve	New Harryburgh	North Ca...	Lithu...	1
499	499	Sherry Ander...	hthomas@yahoo.com	2745644948	2572 Brown Drive Apt. ...	Greenmouth	Nevada	Brazil	1
500	500	Tammy Willia...	uvasquez@gmail.com	513-422-8499	089 Robert Club	Port Tyler	Connecti...	Tuvalu	1
501	10001	John A. Doe	john.adam.doe@example....	987-654-3210	456 Oak Street	Shelbyville	MO	USA	1

8. DeleteCustomer

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[DeleteCustomer]    Script Date: 1/4/2025 12:38:31 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[DeleteCustomer]
    @CustomerID INT
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the specified CustomerID exists.
        IF NOT EXISTS (SELECT 1 FROM Customer WHERE CustomerID = @CustomerID)
        BEGIN
            THROW 50008, 'CustomerID not found. Delete operation aborted.', 1;
        END;

        -- Perform the delete operation.
        DELETE FROM Customer WHERE CustomerID = @CustomerID;

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
```



```
-- Customer Procedures
SELECT * FROM Customer;
--
EXEC InsertCustomer
    @CustomerID = 10001,
    @CustomerName = 'John Doe',
    @Email = 'johndoe@example.com',
    @TelephoneNo = '123-456-7890',
    @Street = '123 Elm Street',
    @City = 'Springfield',
    @State = 'IL',
    @Country = 'USA',
    @CompanyID = 1;

SELECT * FROM Customer;
```

100 %

Results Messages

	CustomerID	CustomerName	Email	TelephoneNo	Street	City	State	Country	CompanyID
483	483	Heidi Ferguson	alexanderholmes@yahoo...	001-765-213-28...	7112 Timothy Inlet	Donmouth	Rhode Is...	Maldi...	1
484	484	Gloria Mckinn...	walkermelissa@watts.com	216.110.7603	89071 Williams Hollow	Port Colinvill	Rhode Is...	Costa...	1
485	485	Christina Lam...	thomasjulia@rush-bailey.c...	938.911.9083x3...	87592 Nunez Coves	Port Michael	Missouri	Jamai...	1
486	486	Darren Rodri...	owensapril@yahoo.com	001-548-015-71...	6317 White Spring Suit...	North Douglasfurt	Vermont	Namibia	1
487	487	Natalie Stafford	coojames@larson.com	674.994.6579	8427 David Overpass ...	Lake Brianland	New York	Philip...	1
488	488	Christopher J...	kevinhanson@gmail.com	+1-877-347-784...	67567 Johnson Fall Sui...	West Jennifer	New York	Serbia	1
489	489	Michael Keller	khart@hotmail.com	6479695450	025 Pacheco Prairie	Port Patriciamouth	Missouri	Bhutan	1
490	490	William Jenkins	johnwong@miller.com	208-507-6986x4...	9659 Lindsey Courts A...	Garrettport	Mississippi	Jordan	1
491	491	Mark Dickson	karlwilliams@morales.com	001-919-541-98...	468 Gonzalez Looks Su...	North Courtney	North Da...	Tokelau	1
492	492	Jonathan Gu...	sharon89@yahoo.com	001-083-572-3246	1364 Lucas Glens Apt. ...	East Anthony	Maryland	Kyrgy...	1
493	493	Dr. Javier Bul...	josephjohnson@hopkins-t...	001-700-159-41...	818 Crawford Course	New David	Massach...	Gree...	1
494	494	Nancy Shaffer	grobinsont@lewis.info	(490)785-1586x...	540 McKenzie Shore A...	Parkerstad	Utah	British...	1
495	495	Cynthia Ramos	qpennington@davis-carso...	327-560-4177x6...	481 Fisher Stream	Shawntown	Wisconsin	Austria	1
496	496	Catherine Ra...	moralescourtney@johnsto...	296.577.2429x5...	66930 Sandra Rue Apt....	Jacquelineburgh	Arkansas	Iran	1
497	497	Lindsay Hern...	staceywilson@hotmail.com	677.133.3447x6...	3626 Anthony Greens	New Shannon	Kansas	Heard...	1
498	498	Maria Miller	monicamccann@yahoo.com	+1-366-941-180...	48964 Lambert Curve	New Harryburgh	North Ca...	Lithu...	1
499	499	Sherry Ander...	hthomas@yahoo.com	2745644948	2572 Brown Drive Apt. ...	Greenmouth	Nevada	Brazil	1
500	500	Tammy Willia...	uvasquez@gmail.com	513-422-8499	089 Robert Club	Port Tyler	Connecti...	Tuvalu	1
501	10001	John Doe	johndoe@example.com	123-456-7890	123 Elm Street	Springfield	IL	USA	1

Query executed successfully.

localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 501 rows

```
-- Delete
SELECT * FROM Customer;

EXEC DeleteCustomer
    @CustomerID = 10001;

SELECT * FROM Customer;
```

100 %

Results Messages

	CustomerID	CustomerName	Email	TelephoneNo	Street	City	State	Country	CompanyI
482	482	Jose Johnston	bonniethomas@yahoo.com	001-252-045-65...	219 Amy Well	Margaretport	Louisiana	Poland	1
483	483	Heidi Ferguson	alexanderholmes@yahoo....	001-765-213-28...	7112 Timothy Inlet	Donmouth	Rhode Is...	Maldi...	1
484	484	Gloria Mokinn...	walkemelissa@watts.com	216.110.7603	89071 Williams Hollow	Port Colinvill	Rhode Is...	Costa...	1
485	485	Christina Lam...	thomasjulie@rush-bailey.c...	938.911.9083x3...	87592 Nunez Coves	Port Michael	Missouri	Jamai...	1
486	486	Darren Rodri...	owensapril@yahoo.com	001-548-015-71...	6317 White Spring Suit...	North Douglasfurt	Vermont	Namibia	1
487	487	Natalie Stafford	coxjames@larson.com	674.994.6579	8427 David Overpass ...	Lake Brianland	New York	Philp...	1
488	488	Christopher J...	kevinhanson@gmail.com	+1-877-347-784...	67567 Johnson Fall Sui...	West Jennifer	New York	Serbia	1
489	489	Michael Keller	khant@hotmail.com	6479695450	025 Pacheco Prairie	Port Patriciamouth	Missouri	Bhutan	1
490	490	William Jenkins	johnwong@miller.com	208-507-6986x4...	9659 Lindsey Courts A...	Garrettport	Mississippi	Jordan	1
491	491	Mark Dickson	karlwilliams@morales.com	001-919-541-98...	468 Gonzalez Looks Su...	North Courtney	North Da...	Tokelau	1
492	492	Jonathan Gu...	sharon89@yahoo.com	001-083-572-3246	1364 Lucas Glens Apt. ...	East Anthony	Maryland	Kyrgy...	1
493	493	Dr. Javier Bul...	josephjohnson@hopkins-t...	001-700-159-41...	818 Crawford Course	New David	Massach...	Gree...	1
494	494	Nancy Shaffer	grobinson@lewis.info	(490)785-1586x...	540 Mckenzie Shore A...	Parkerstad	Utah	British...	1
495	495	Cynthia Ramos	qpennington@davis-carso...	327-560-4177x6...	481 Fisher Stream	Shawntown	Wisconsin	Austria	1
496	496	Catherine Ra...	moralescourtney@johnsto...	296.577.2429x5...	66930 Sandra Rue Apt....	Jacquelineburgh	Arkansas	Iran	1
497	497	Lindsay Hern...	staceywilson@hotmail.com	677.133.3447x6...	3626 Anthony Greens	New Shannon	Kansas	Heard...	1
498	498	Maria Miller	monicamocann@yahoo.com	+1-366-941-180...	48964 Lambert Curve	New Harryburgh	North Ca...	Lithu...	1
499	499	Sherry Ander...	hthomas@yahoo.com	2745644948	2572 Brown Drive Apt. ...	Greenmouth	Nevada	Brazil	1
500	500	Tammy Willia...	uvasquez@gmail.com	513-422-8499	089 Robert Club	Port Tyler	Connecti...	Tuvalu	1

✓ Query executed successfully.

localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 500 rows

9. InsertDepartment

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[InsertDepartment]    Script Date: 1/4/2025 1:00:06 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-----
CREATE PROCEDURE [dbo].[InsertDepartment]
    @DepartmentNo INT,           -- Unique identifier for the department
    @DepartmentName VARCHAR(50), -- Name of the department
    @NoOfEmployees INT,         -- Number of employees in the department
    @Mission VARCHAR(100),      -- Mission statement of the department
    @CompanyID INT              -- ID of the company the department belongs to
AS
BEGIN
    -- Start a database transaction to ensure atomicity.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the DepartmentNo already exists in the table.
        IF EXISTS (SELECT 1 FROM Department WHERE DepartmentNo = @DepartmentNo)
        BEGIN
            THROW 51001, 'DepartmentNo already exists. Insert operation aborted.', 1;
        END;

        -- Insert a new record into the Department table.
        INSERT INTO Department (DepartmentNo, DepartmentName, NoOfEmployees, Mission, CompanyID)
        VALUES (@DepartmentNo, @DepartmentName, @NoOfEmployees, @Mission, @CompanyID);

        -- Commit the transaction after successful insertion.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of any errors.
        ROLLBACK TRANSACTION;
        THROW; -- Re-throw the caught error for the caller to handle.
    END CATCH;
END;
GO
```

```
SELECT * FROM Customer;
```

```
-- Department Procedures
```

```
-- Insert
```

```
SELECT * FROM Department;
```

100 %

Results Messages

	DepartmentNo	DepartmentName	NoOfEmployees	Mission	CompanyID
1	1	Administration	10	To oversee daily operations and coordination	1
2	2	Cloud Services	50	To provide reliable cloud-based solutions and supp...	1
3	3	Network Management	40	To maintain and optimize network infrastructure	1
4	4	Software Repair	30	To troubleshoot and fix software-related issues	1
5	5	Hardware Repair	25	To repair and maintain hardware devices	1
6	6	Security Systems	35	To ensure robust security and protection against th...	1
7	7	Software Development	45	To design and build innovative software solutions	1
8	8	Data Analytics	20	To analyze data for insights and decision-making	1
9	9	Customer Support	50	To assist customers with their technical and service...	1
10	10	Testing and Quality Assurance	25	To test and ensure the quality of services and prod...	1
11	11	Research and Development	15	To innovate and develop next-generation technolo...	1
12	12	Human Resources	10	To manage and support the company's workforce	1
13	13	Finance and Accounting	5	To handle financial transactions and reporting	1
14	14	Marketing and Sales	15	To promote and sell the company's products and s...	1
15	15	Legal and Compliance	5	To ensure legal compliance and protect the compa...	1

```
@CompanyID = 1;
```

```
SELECT * FROM Department;
```

100 %

Results Messages

	DepartmentNo	DepartmentName	NoOfEmployees	Mission	CompanyID
1	1	Administration	10	To oversee daily operations and coordination	1
2	2	Cloud Services	50	To provide reliable cloud-based solutions and supp...	1
3	3	Network Management	40	To maintain and optimize network infrastructure	1
4	4	Software Repair	30	To troubleshoot and fix software-related issues	1
5	5	Hardware Repair	25	To repair and maintain hardware devices	1
6	6	Security Systems	35	To ensure robust security and protection against th...	1
7	7	Software Development	45	To design and build innovative software solutions	1
8	8	Data Analytics	20	To analyze data for insights and decision-making	1
9	9	Customer Support	50	To assist customers with their technical and service...	1
10	10	Testing and Quality Assurance	25	To test and ensure the quality of services and prod...	1
11	11	Research and Development	15	To innovate and develop next-generation technolo...	1
12	12	Human Resources	10	To manage and support the company's workforce	1
13	13	Finance and Accounting	5	To handle financial transactions and reporting	1
14	14	Marketing and Sales	15	To promote and sell the company's products and s...	1
15	15	Legal and Compliance	5	To ensure legal compliance and protect the compa...	1
16	101	Engineering	50	Innovate technology solutions.	1

10. GetDepartment

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[GetDepartment]    Script Date: 1/4/2025 1:04:24 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-----
CREATE PROCEDURE [dbo].[GetDepartment]
    @DepartmentNo INT = NULL -- Optional parameter to fetch specific department details
AS
BEGIN
    BEGIN TRY
        -- If no DepartmentNo is provided, fetch all departments.
        IF @DepartmentNo IS NULL
        BEGIN
            SELECT * FROM Department;
        END
        ELSE
        BEGIN
            -- Validate if the provided DepartmentNo exists.
            IF NOT EXISTS (SELECT 1 FROM Department WHERE DepartmentNo = @DepartmentNo)
            BEGIN
                THROW 51002, 'DepartmentNo not found. Read operation aborted.', 1;
            END;

            -- Fetch the details of the specified department.
            SELECT * FROM Department WHERE DepartmentNo = @DepartmentNo;
        END;
    END TRY
    BEGIN CATCH
        -- Re-throw the caught error for the caller to handle.
        THROW;
    END CATCH;
END;
GO

-----
```

-- Get

```
EXEC GetDepartment
    @DepartmentNo = 10;
```

100 %

Results Messages

	DepartmentNo	DepartmentName	NoOfEmployees	Mission	CompanyID
1	10	Testing and Quality Assurance	25	To test and ensure the quality of services and p...	1

```
-- Get
```

```
EXEC GetDepartment  
    @DepartmentNo = 10;
```

```
EXEC GetDepartment;
```

```
SELECT * FROM Department;
```

100 %

Results Messages

	DepartmentNo	DepartmentName	NoOfEmployees	Mission	CompanyID
1	1	Administration	10	To oversee daily operations and coordination	1
2	2	Cloud Services	50	To provide reliable cloud-based solutions and supp...	1
3	3	Network Management	40	To maintain and optimize network infrastructure	1
4	4	Software Repair	30	To troubleshoot and fix software-related issues	1
5	5	Hardware Repair	25	To repair and maintain hardware devices	1
6	6	Security Systems	35	To ensure robust security and protection against th...	1
7	7	Software Development	45	To design and build innovative software solutions	1
8	8	Data Analytics	20	To analyze data for insights and decision-making	1
9	9	Customer Support	50	To assist customers with their technical and service...	1
10	10	Testing and Quality Assurance	25	To test and ensure the quality of services and prod...	1
11	11	Research and Development	15	To innovate and develop next-generation technolo...	1
12	12	Human Resources	10	To manage and support the company's workforce	1
13	13	Finance and Accounting	5	To handle financial transactions and reporting	1
14	14	Marketing and Sales	15	To promote and sell the company's products and s...	1
15	15	Legal and Compliance	5	To ensure legal compliance and protect the compa...	1

11. UpdateDepartment

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[UpdateDepartment]    Script Date: 1/4/2025 1:17:24 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-----
CREATE PROCEDURE [dbo].[UpdateDepartment]
    @DepartmentNo INT,
    @DepartmentName VARCHAR(50) = NULL,
    @NoOfEmployees INT = NULL,
    @Mission VARCHAR(100) = NULL,
    @CompanyID INT = NULL
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the specified DepartmentNo exists in the Department table.
        IF NOT EXISTS (SELECT 1 FROM Department WHERE DepartmentNo = @DepartmentNo)
        BEGIN
            THROW 51003, 'DepartmentNo not found. Update operation aborted.', 1;
        END;

        -- Perform the update operation, using existing values if no new values are provided.
        UPDATE Department
        SET
            DepartmentName = ISNULL(@DepartmentName, DepartmentName),
            NoOfEmployees = ISNULL(@NoOfEmployees, NoOfEmployees),
            Mission = ISNULL(@Mission, Mission),
            CompanyID = ISNULL(@CompanyID, CompanyID)
        WHERE DepartmentNo = @DepartmentNo;

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
-----
```

```
-- Update
SELECT * FROM Department;
```

```
EXEC InsertDepartment
    @DepartmentNo = 101,
    @DepartmentName = 'Engineering',
    @NoOfEmployees = 50,
    @Mission = 'Innovate technology solutions.',
    @CompanyID = 1;
SELECT * FROM Department;

EXEC UpdateDepartment
    @DepartmentNo = 101,
    @DepartmentName = 'Engineering Department',
    @NoOfEmployees = 60,
    @Mission = 'Innovate technology solutions for a better future.',
    @CompanyID = 1;
```

100 %

Results Messages

	DepartmentNo	DepartmentName	NoOfEmployees	Mission	CompanyID
1	1	Administration	10	To oversee daily operations and coordination	1
2	2	Cloud Services	50	To provide reliable cloud-based solutions and supp...	1
3	3	Network Management	40	To maintain and optimize network infrastructure	1
4	4	Software Repair	30	To troubleshoot and fix software-related issues	1
5	5	Hardware Repair	25	To repair and maintain hardware devices	1
6	6	Security Systems	35	To ensure robust security and protection against th...	1
7	7	Software Development	45	To design and build innovative software solutions	1
8	8	Data Analytics	20	To analyze data for insights and decision-making	1
9	9	Customer Support	50	To assist customers with their technical and service...	1
10	10	Testing and Quality Assurance	25	To test and ensure the quality of services and prod...	1
11	11	Research and Development	15	To innovate and develop next-generation technolo...	1
12	12	Human Resources	10	To manage and support the company's workforce	1
13	13	Finance and Accounting	5	To handle financial transactions and reporting	1
14	14	Marketing and Sales	15	To promote and sell the company's products and s...	1
15	15	Legal and Compliance	5	To ensure legal compliance and protect the compa...	1
16	101	Engineering	50	Innovate technology solutions.	1


```
-- Update
SELECT * FROM Department;

EXEC InsertDepartment
    @DepartmentNo = 101,
    @DepartmentName = 'Engineering',
    @NoOfEmployees = 50,
    @Mission = 'Innovate technology solutions.',
    @CompanyID = 1;

SELECT * FROM Department;

EXEC UpdateDepartment
    @DepartmentNo = 101,
    @DepartmentName = 'Engineering Department',
    @NoOfEmployees = 60,
    @Mission = 'Innovate technology solutions for a better future.',
    @CompanyID = 1;

SELECT * FROM Department;
```

100 %

Results Messages

	DepartmentNo	DepartmentName	NoOfEmployees	Mission	CompanyID
1	1	Administration	10	To oversee daily operations and coordination	1
2	2	Cloud Services	50	To provide reliable cloud-based solutions and supp...	1
3	3	Network Management	40	To maintain and optimize network infrastructure	1
4	4	Software Repair	30	To troubleshoot and fix software-related issues	1
5	5	Hardware Repair	25	To repair and maintain hardware devices	1
6	6	Security Systems	35	To ensure robust security and protection against th...	1
7	7	Software Development	45	To design and build innovative software solutions	1
8	8	Data Analytics	20	To analyze data for insights and decision-making	1
9	9	Customer Support	50	To assist customers with their technical and service...	1
10	10	Testing and Quality Assurance	25	To test and ensure the quality of services and prod...	1
11	11	Research and Development	15	To innovate and develop next-generation technolo...	1
12	12	Human Resources	10	To manage and support the company's workforce	1
13	13	Finance and Accounting	5	To handle financial transactions and reporting	1
14	14	Marketing and Sales	15	To promote and sell the company's products and s...	1
15	15	Legal and Compliance	5	To ensure legal compliance and protect the compa...	1
16	101	Engineering Department	60	Innovate technology solutions for a better future.	1

12. DeleteDepartment

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[DeleteDepartment]    Script Date: 1/4/2025 1:02:26 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-----
CREATE PROCEDURE [dbo].[DeleteDepartment]
    @DepartmentNo INT
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the specified DepartmentNo exists in the Department table.
        IF NOT EXISTS (SELECT 1 FROM Department WHERE DepartmentNo = @DepartmentNo)
        BEGIN
            THROW 51004, 'DepartmentNo not found. Delete operation aborted.', 1;
        END;

        -- Perform the delete operation.
        DELETE FROM Department WHERE DepartmentNo = @DepartmentNo;

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO

-----
```

```
@CompanyID = 1;
```

```
SELECT * FROM Department;
```

100 %

Results Messages

	DepartmentNo	DepartmentName	NoOfEmployees	Mission	CompanyID
1	1	Administration	10	To oversee daily operations and coordination	1
2	2	Cloud Services	50	To provide reliable cloud-based solutions and supp...	1
3	3	Network Management	40	To maintain and optimize network infrastructure	1
4	4	Software Repair	30	To troubleshoot and fix software-related issues	1
5	5	Hardware Repair	25	To repair and maintain hardware devices	1
6	6	Security Systems	35	To ensure robust security and protection against th...	1
7	7	Software Development	45	To design and build innovative software solutions	1
8	8	Data Analytics	20	To analyze data for insights and decision-making	1
9	9	Customer Support	50	To assist customers with their technical and service...	1
10	10	Testing and Quality Assurance	25	To test and ensure the quality of services and prod...	1
11	11	Research and Development	15	To innovate and develop next-generation technolo...	1
12	12	Human Resources	10	To manage and support the company's workforce	1
13	13	Finance and Accounting	5	To handle financial transactions and reporting	1
14	14	Marketing and Sales	15	To promote and sell the company's products and s...	1
15	15	Legal and Compliance	5	To ensure legal compliance and protect the compa...	1
16	101	Engineering	50	Innovate technology solutions.	1

```
-- Delete
```

```
SELECT * FROM Department;
```

```
EXEC DeleteDepartment  
    @DepartmentNo = 101;
```

```
SELECT * FROM Department;
```

100 %

Results Messages

	DepartmentNo	DepartmentName	NoOfEmployees	Mission	CompanyID
1	1	Administration	10	To oversee daily operations and coordination	1
2	2	Cloud Services	50	To provide reliable cloud-based solutions and supp...	1
3	3	Network Management	40	To maintain and optimize network infrastructure	1
4	4	Software Repair	30	To troubleshoot and fix software-related issues	1
5	5	Hardware Repair	25	To repair and maintain hardware devices	1
6	6	Security Systems	35	To ensure robust security and protection against th...	1
7	7	Software Development	45	To design and build innovative software solutions	1
8	8	Data Analytics	20	To analyze data for insights and decision-making	1
9	9	Customer Support	50	To assist customers with their technical and service...	1
10	10	Testing and Quality Assurance	25	To test and ensure the quality of services and prod...	1
11	11	Research and Development	15	To innovate and develop next-generation technolo...	1
12	12	Human Resources	10	To manage and support the company's workforce	1
13	13	Finance and Accounting	5	To handle financial transactions and reporting	1
14	14	Marketing and Sales	15	To promote and sell the company's products and s...	1
15	15	Legal and Compliance	5	To ensure legal compliance and protect the compa...	1

13. InsertItem

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

```
USE [Database-001]
GO

/***** Object:  StoredProcedure [dbo].[InsertItem]    Script Date: 1/4/2025 1:28:34 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[InsertItem]
    @ItemID INT,
    @ItemName VARCHAR(50),
    @Quantity INT,
    @Price INT,
    @CompanyID INT
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the ItemID already exists in the Item table.
        IF EXISTS (SELECT 1 FROM Item WHERE ItemID = @ItemID)
        BEGIN
            THROW 52001, 'ItemID already exists. Insert operation aborted.', 1;
        END;

        -- Perform the insert operation.
        INSERT INTO Item (ItemID, ItemName, Quantity, Price, CompanyID)
        VALUES (@ItemID, @ItemName, @Quantity, @Price, @CompanyID);

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
```

```

-- Items Procedures
-- Insert
SELECT * FROM Item;

EXEC InsertItem
    @ItemID = 100,
    @ItemName = 'Laptop',
    @Quantity = 50,
    @Price = 1000,
    @CompanyID = 1;

SELECT * FROM Item;

-- Delete
SELECT * FROM Item;

```

100 %

Results Messages

	ItemID	ItemName	Quantity	Price	CompanyID
11	11	Webcam	35	45	1
12	12	Smartwatch	22	200	1
13	13	Graphic Tablet	8	350	1
14	14	Speaker	28	70	1
15	15	Power Bank	55	25	1
16	16	USB Flash Drive	100	10	1
17	17	Desk Chair	10	120	1
18	18	Office Desk	7	300	1
19	19	Gaming Chair	5	250	1
20	20	Projector	6	400	1
21	21	Camera	12	550	1
22	22	Tripod	20	60	1
23	23	Whiteboard	15	75	1
24	24	Smart TV	8	700	1
25	25	Air Conditioner	5	1000	1
26	26	Microwave	10	180	1
27	27	Washing Machine	3	800	1
28	28	Refrigerator	4	1500	1
29	29	Dishwasher	3	900	1
30	30	Electric Kettle	50	35	1

✓ Query executed successfully.

localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 30 rows

```
EXEC InsertItem
    @ItemID = 100,
    @ItemName = 'Laptop',
    @Quantity = 50,
    @Price = 1000,
    @CompanyID = 1;
```

```
SELECT * FROM Item;
```

```
-- Delete
```

```
SELECT * FROM Item;
```

100 %

Results Messages

	ItemID	ItemName	Quantity	Price	CompanyID
12	12	Smartwatch	22	200	1
13	13	Graphic Tablet	8	350	1
14	14	Speaker	28	70	1
15	15	Power Bank	55	25	1
16	16	USB Flash Drive	100	10	1
17	17	Desk Chair	10	120	1
18	18	Office Desk	7	300	1
19	19	Gaming Chair	5	250	1
20	20	Projector	6	400	1
21	21	Camera	12	550	1
22	22	Tripod	20	60	1
23	23	Whiteboard	15	75	1
24	24	Smart TV	8	700	1
25	25	Air Conditioner	5	1000	1
26	26	Microwave	10	180	1
27	27	Washing Machine	3	800	1
28	28	Refrigerator	4	1500	1
29	29	Dishwasher	3	900	1
30	30	Electric Kettle	50	35	1
31	100	Laptop	50	1000	1

✓ Query executed successfully.

localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 31 rows

14. GetItem

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[GetItem]    Script Date: 1/4/2025 1:32:31 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[GetItem]
    @ItemID INT = NULL
AS
BEGIN
    BEGIN TRY
        -- If no ItemID is provided, fetch all items.
        IF @ItemID IS NULL
        BEGIN
            SELECT * FROM Item;
        END
        ELSE
        BEGIN
            -- Check if the specified ItemID exists.
            IF NOT EXISTS (SELECT 1 FROM Item WHERE ItemID = @ItemID)
            BEGIN
                THROW 52002, 'ItemID not found. Read operation aborted.', 1;
            END;

            -- Retrieve details of the specified item.
            SELECT * FROM Item WHERE ItemID = @ItemID;
        END;
    END TRY
    BEGIN CATCH
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
```

```
-- Get
SELECT * FROM Item;
-- Retrieve all items:
EXEC GetItem;

-- Retrieve a specific item:
EXEC GetItem @ItemID = 1;

-- Update
SELECT * FROM Item;
```

100 %

Results Messages

	ItemID	ItemName	Quantity	Price	CompanyID
1	1	Laptop	10	1200	1
2	2	Smartphone	20	800	1
3	3	Monitor	15	300	1
4	4	Keyboard	50	20	1
5	5	Mouse	60	15	1
6	6	Printer	12	150	1
7	7	Router	25	75	1
8	8	External Hard Drive	18	90	1
9	9	Tablet	30	250	1
10	10	Headphones	40	50	1
11	11	Webcam	35	45	1
12	12	Smartwatch	22	200	1
13	13	Graphic Tablet	8	350	1
14	14	Speaker	28	70	1
15	15	Power Bank	55	25	1
16	16	USB Flash Drive	100	10	1
17	17	Desk Chair	10	120	1
18	18	Office Desk	7	300	1
19	19	Gaming Chair	5	250	1
20	20	Projector	6	400	1

✓ Query executed successfully. | localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 30 rows

```
-- Retrieve a specific item:
EXEC GetItem @ItemID = 1;

-- Update
SELECT * FROM Item;
```

100 %

Results Messages

	ItemID	ItemName	Quantity	Price	CompanyID
1	1	Laptop	10	1200	1

15. UpdateItem

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[UpdateItem]    Script Date: 1/4/2025 1:36:06 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[UpdateItem]
    @ItemID INT,
    @ItemName VARCHAR(50) = NULL,
    @Quantity INT = NULL,
    @Price INT = NULL,
    @CompanyID INT = NULL
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the specified ItemID exists in the Item table.
        IF NOT EXISTS (SELECT 1 FROM Item WHERE ItemID = @ItemID)
        BEGIN
            THROW 52003, 'ItemID not found. Update operation aborted.', 1;
        END;

        -- Perform the update operation, using existing values if no new values are provided.
        UPDATE Item
        SET
            ItemName = ISNULL(@ItemName, ItemName),
            Quantity = ISNULL(@Quantity, Quantity),
            Price = ISNULL(@Price, Price),
            CompanyID = ISNULL(@CompanyID, CompanyID)
        WHERE ItemID = @ItemID;

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
```

```
SELECT * FROM Item;
```

```
EXEC UpdateItem  
    @ItemID = 100,  
    @ItemName = 'Gaming Laptop',  
    @Price = 1200;  
  
SELECT * FROM Item;
```

100 %

Results Messages

	ItemID	ItemName	Quantity	Price	CompanyID
12	12	Smartwatch	22	200	1
13	13	Graphic Tablet	8	350	1
14	14	Speaker	28	70	1
15	15	Power Bank	55	25	1
16	16	USB Flash Drive	100	10	1
17	17	Desk Chair	10	120	1
18	18	Office Desk	7	300	1
19	19	Gaming Chair	5	250	1
20	20	Projector	6	400	1
21	21	Camera	12	550	1
22	22	Tripod	20	60	1
23	23	Whiteboard	15	75	1
24	24	Smart TV	8	700	1
25	25	Air Conditioner	5	1000	1
26	26	Microwave	10	180	1
27	27	Washing Machine	3	800	1
28	28	Refrigerator	4	1500	1
29	29	Dishwasher	3	900	1
30	30	Electric Kettle	50	35	1
31	100	Laptop	50	1000	1

Query executed successfully.

localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 31 rows

```
EXEC UpdateItem
    @ItemID = 100,
    @ItemName = 'Gaming Laptop',
    @Price = 1200;
```

```
SELECT * FROM Item;
```

100 %

Results Messages

	ItemID	ItemName	Quantity	Price	CompanyID
12	12	Smartwatch	22	200	1
13	13	Graphic Tablet	8	350	1
14	14	Speaker	28	70	1
15	15	Power Bank	55	25	1
16	16	USB Flash Drive	100	10	1
17	17	Desk Chair	10	120	1
18	18	Office Desk	7	300	1
19	19	Gaming Chair	5	250	1
20	20	Projector	6	400	1
21	21	Camera	12	550	1
22	22	Tripod	20	60	1
23	23	Whiteboard	15	75	1
24	24	Smart TV	8	700	1
25	25	Air Conditioner	5	1000	1
26	26	Microwave	10	180	1
27	27	Washing Machine	3	800	1
28	28	Refrigerator	4	1500	1
29	29	Dishwasher	3	900	1
30	30	Electric Kettle	50	35	1
31	100	Gaming Laptop	50	1200	1

✓ Query executed successfully.

localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 31 rows

16. DeleteItem

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[DeleteItem]    Script Date: 1/4/2025 1:30:40 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[DeleteItem]
    @ItemID INT
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the specified ItemID exists in the Item table.
        IF NOT EXISTS (SELECT 1 FROM Item WHERE ItemID = @ItemID)
        BEGIN
            THROW 52004, 'ItemID not found. Delete operation aborted.', 1;
        END;

        -- Perform the delete operation.
        DELETE FROM Item WHERE ItemID = @ItemID;

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
```

```
EXEC InsertItem
    @ItemID = 100,
    @ItemName = 'Laptop',
    @Quantity = 50,
    @Price = 1000,
    @CompanyID = 1;
```

```
SELECT * FROM Item;
```

```
-- Delete
```

```
SELECT * FROM Item;
```

100 %

Results Messages

	ItemID	ItemName	Quantity	Price	CompanyID
12	12	Smartwatch	22	200	1
13	13	Graphic Tablet	8	350	1
14	14	Speaker	28	70	1
15	15	Power Bank	55	25	1
16	16	USB Flash Drive	100	10	1
17	17	Desk Chair	10	120	1
18	18	Office Desk	7	300	1
19	19	Gaming Chair	5	250	1
20	20	Projector	6	400	1
21	21	Camera	12	550	1
22	22	Tripod	20	60	1
23	23	Whiteboard	15	75	1
24	24	Smart TV	8	700	1
25	25	Air Conditioner	5	1000	1
26	26	Microwave	10	180	1
27	27	Washing Machine	3	800	1
28	28	Refrigerator	4	1500	1
29	29	Dishwasher	3	900	1
30	30	Electric Kettle	50	35	1
31	100	Laptop	50	1000	1

✓ Query executed successfully.

localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 31 rows

```
-- Delete
SELECT * FROM Item;
EXEC DeleteItem @ItemID = 100;
SELECT * FROM Item;

-- Get
SELECT * FROM Item;
```

100 %

Results Messages

	ItemID	ItemName	Quantity	Price	CompanyID
11	11	Webcam	35	45	1
12	12	Smartwatch	22	200	1
13	13	Graphic Tablet	8	350	1
14	14	Speaker	28	70	1
15	15	Power Bank	55	25	1
16	16	USB Flash Drive	100	10	1
17	17	Desk Chair	10	120	1
18	18	Office Desk	7	300	1
19	19	Gaming Chair	5	250	1
20	20	Projector	6	400	1
21	21	Camera	12	550	1
22	22	Tripod	20	60	1
23	23	Whiteboard	15	75	1
24	24	Smart TV	8	700	1
25	25	Air Conditioner	5	1000	1
26	26	Microwave	10	180	1
27	27	Washing Machine	3	800	1
28	28	Refrigerator	4	1500	1
29	29	Dishwasher	3	900	1
30	30	Electric Kettle	50	35	1

✓ Query executed successfully.

localhost\MSSQLSERVER01 (16... | PIRATE-960\Pirate-960 ... | Database-001 | 00:00:00 | 30 rows

17. InsertService

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[InsertService]    Script Date: 1/4/2025 1:25:31 PM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- =====
-- *****
-- PROCEDURES FOR SERVICE TABLE CRUD OPERATIONS
-- These procedures perform Create, Read, Update, and Delete (CRUD) operations
-- on the `Service` table with appropriate error handling and transactional support.
-- *****
-- =====

-- InsertService: Inserts a new service into the Service table.
-- Parameters:
-- @ServiceID (INT): Unique identifier for the service.
-- @Name (VARCHAR(50)): Name of the service.
-- @StartDate (DATE): Start date of the service.
-- @EndDate (DATE): End date of the service.
-- @Price (INT): Price of the service.
-- @Status (VARCHAR(50)): Status of the service.
-- @City (VARCHAR(50)): City where the service is provided.
-- @State (VARCHAR(50)): State where the service is provided.
-- @Country (VARCHAR(50)): Country where the service is provided.
-- @Cloud (VARCHAR(50)): Cloud service type (optional).
-- @Security (VARCHAR(50)): Security service type (optional).
-- @Repair (VARCHAR(50)): Repair service type (optional).
-- @Network (VARCHAR(50)): Network service type (optional).
-- @SoftwareDev (VARCHAR(50)): Software development service type (optional).
-- @CompanyID (INT): ID of the company providing the service.
-- @DepartmentNo (INT): Department number associated with the service.
-- =====

CREATE PROCEDURE [dbo].[InsertService]
    @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) = NULL,
    @Security VARCHAR(50) = NULL,
    @Repair VARCHAR(50) = NULL,
    @Network VARCHAR(50) = NULL,
```

100 %

Connected. (1/1)

```

-- @DepartmentNo (INT): Department number associated with the service.
-----
CREATE PROCEDURE [dbo].[InsertService]
    @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) = NULL,
    @Security VARCHAR(50) = NULL,
    @Repair VARCHAR(50) = NULL,
    @Network VARCHAR(50) = NULL,
    @SoftwareDev VARCHAR(50) = NULL,
    @CompanyID INT,
    @DepartmentNo INT
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the ServiceID already exists in the Service table.
        IF EXISTS (SELECT 1 FROM Service WHERE ServiceID = @ServiceID)
        BEGIN
            THROW 53001, 'ServiceID already exists. Insert operation aborted.', 1;
        END;

        -- Perform the insert operation.
        INSERT INTO Service (
            ServiceID, Name, StartDate, EndDate, Price, Status, City, State, Country, Cloud, Security, Repair, Network, SoftwareDev, CompanyID, DepartmentNo
        )
        VALUES (
            @ServiceID, @Name, @StartDate, @EndDate, @Price, @Status, @City, @State, @Country, @Cloud, @Security, @Repair, @Network, @SoftwareDev, @CompanyID, @DepartmentNo
        );

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO

```

100 %

Connected. (1/1)

localhost\MSSQLSERVER01 (16... | PIRATE-


```
-- Service Procedures
-- Insert
SELECT * FROM Service;

EXEC InsertService
@ServiceID = 1,
@Name = 'Cloud Hosting',
@StartDate = '2024-01-01',
@EndDate = '2024-12-31',
@Price = 5000,
@Status = 'Active',
@City = 'New York',
@State = 'NY',
@Country = 'USA',
@Cloud = 'AWS',
@CompanyID = 1,
@DepartmentNo = 101;

SELECT * FROM Service;
```

100 %

Results Messages

	ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo	
1	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
2	2002	Cloud Backup Services	2023-02-01	2023-02-13	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	12	1	2	
3	2003	Cloud Storage	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
4	2004	Cloud Application Hosting	2023-04-01	2023-04-13	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	12	1	2	
5	2005	Cloud Security Solutions	2023-05-01	2023-05-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
6	2006	Private Cloud Solutions	2023-06-01	2023-06-14	4500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
7	2007	Cloud Platform Development	2023-07-01	2023-07-14	6000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
8	2008	Disaster Recovery Cloud	2023-08-01	2023-08-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
9	2009	Cloud Data Analytics	2023-09-01	2023-09-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
10	2010	Cloud DevOps Services	2023-10-01	2023-10-14	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
11	2011	Cloud Migration Solutions	2023-11-01	2023-11-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
12	2012	Cloud API Integration	2023-12-01	2023-12-14	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
13	2013	Hybrid Cloud Solutions	2023-01-15	2023-01-29	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	14	1	2	
14	2014	Cloud Infrastructure Mana...	2023-02-15	2023-02-28	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
15	2015	Multi-Cloud Solutions	2023-03-15	2023-03-28	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
16	2016	Cloud Computing Solutions	2023-04-15	2023-04-28	6000	Active	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
17	2017	Cloud Database Solutions	2023-05-15	2023-05-28	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
18	2018	Cloud Storage for Enterpris...	2023-06-15	2023-06-28	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
19	2019	Cloud-Based Business Sol...	2023-07-15	2023-07-28	5000	Active	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
20	2020	Cloud Disaster Recovery	2023-08-15	2023-08-28	6000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
21	2021	Cloud SaaS Solutions	2023-09-15	2023-09-28	5500	Active	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
22	2022	Cloud Virtualization	2023-10-15	2023-10-28	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
23	2023	Cloud Data Storage	2023-11-15	2023-11-28	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2	
24	2024	Cloud Software Solutions	2023-12-15	2023-12-28	5500	Active	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2	

Query executed successfully.

```
-- Service Procedures
-- Insert
SELECT * FROM Service;

EXEC InsertService
    @ServiceID = 1,
    @Name = 'Cloud Hosting',
    @StartDate = '2024-01-01',
    @EndDate = '2024-12-31',
    @Price = 5000,
    @Status = 'Active',
    @City = 'New York',
    @State = 'NY',
    @Country = 'USA',
    @Cloud = 'AWS',
    @CompanyID = 1,
    @DepartmentNo = 101;

SELECT * FROM Service;
```

100 %

Results Messages																	
	ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo
1	1	Cloud Hosting	2024-01-01	2024-12-31	5000	Active	New York	NY	USA	AWS	NULL	NULL	NULL	NULL	365	1	101
2	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
3	2002	Cloud Backup Services	2023-02-01	2023-02-13	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	12	1	2
4	2003	Cloud Storage	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
5	2004	Cloud Application Hosting	2023-04-01	2023-04-13	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	12	1	2
6	2005	Cloud Security Solutions	2023-05-01	2023-05-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
7	2006	Private Cloud Solutions	2023-06-01	2023-06-14	4500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
8	2007	Cloud Platform Development	2023-07-01	2023-07-14	6000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
9	2008	Disaster Recovery Cloud	2023-08-01	2023-08-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
10	2009	Cloud Data Analytics	2023-09-01	2023-09-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
11	2010	Cloud DevOps Services	2023-10-01	2023-10-14	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
12	2011	Cloud Migration Solutions	2023-11-01	2023-11-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
13	2012	Cloud API Integration	2023-12-01	2023-12-14	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
14	2013	Hybrid Cloud Solutions	2023-01-15	2023-01-29	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	14	1	2
15	2014	Cloud Infrastructure Management	2023-02-15	2023-02-28	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
16	2015	Multi-Cloud Solutions	2023-03-15	2023-03-28	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
17	2016	Cloud Computing Solutions	2023-04-15	2023-04-28	6000	Active	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
18	2017	Cloud Database Solutions	2023-05-15	2023-05-28	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
19	2018	Cloud Storage for Enterprises	2023-06-15	2023-06-28	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
20	2019	Cloud-Based Business Solutions	2023-07-15	2023-07-28	5000	Active	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
21	2020	Cloud Disaster Recovery	2023-08-15	2023-08-28	6000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
22	2021	Cloud SaaS Solutions	2023-09-15	2023-09-28	5500	Active	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
23	2022	Cloud Virtualization	2023-10-15	2023-10-28	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
24	2023	Cloud Data Storage	2023-11-15	2023-11-28	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2

Query executed successfully.

```

-- Service Procedures
-- Insert
SELECT * FROM Service;

EXEC InsertService
    @ServiceID = 1,
    @Name = 'Cloud Hosting',
    @StartDate = '2024-01-01',
    @EndDate = '2024-12-31',
    @Price = 5000,
    @Status = 'Active',
    @City = 'New York',
    @State = 'NY',
    @Country = 'USA',
    @Cloud = 'AWS',
    @CompanyID = 1,
    @DepartmentNo = 101;

SELECT * FROM Service;
SELECT * FROM ServiceLog;

```

100 %

Results Messages

	LogID	ServiceID	Action	LogDate
1	5	1	INSERT	2025-01-04 13:24:35.037

18. GetService

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[GetService]    Script Date: 1/4/2025 1:37:14 PM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[GetService]
    @ServiceID INT = NULL
AS
BEGIN
    BEGIN TRY
        -- If no ServiceID is provided, fetch all services.
        IF @ServiceID IS NULL
        BEGIN
            SELECT * FROM Service;
        END
        ELSE
        BEGIN
            -- Check if the specified ServiceID exists.
            IF NOT EXISTS (SELECT 1 FROM Service WHERE ServiceID = @ServiceID)
            BEGIN
                THROW 53002, 'ServiceID not found. Read operation aborted.', 1;
            END;

            -- Retrieve details of the specified service.
            SELECT * FROM Service WHERE ServiceID = @ServiceID;
        END;
    END TRY
    BEGIN CATCH
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
```

```
-- Get
SELECT * FROM Service;

-- Retrieve all services:
EXEC GetService;

-- Retrieve a specific service:
EXEC GetService @ServiceID = 2001;
```

100 %

Results																	
Messages																	
ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo	
1	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
2	2002	Cloud Backup Services	2023-02-01	2023-02-13	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	12	1	2
3	2003	Cloud Storage	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
4	2004	Cloud Application Hosting	2023-04-01	2023-04-13	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	12	1	2
5	2005	Cloud Security Solutions	2023-05-01	2023-05-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
6	2006	Private Cloud Solutions	2023-06-01	2023-06-14	4500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
7	2007	Cloud Platform Development	2023-07-01	2023-07-14	6000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
8	2008	Disaster Recovery Cloud	2023-08-01	2023-08-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
9	2009	Cloud Data Analytics	2023-09-01	2023-09-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
10	2010	Cloud DevOps Services	2023-10-01	2023-10-14	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
11	2011	Cloud Migration Solutions	2023-11-01	2023-11-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
12	2012	Cloud API Integration	2023-12-01	2023-12-14	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
13	2013	Hybrid Cloud Solutions	2023-01-15	2023-01-29	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	14	1	2
ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo	
1	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
2	2002	Cloud Backup Services	2023-02-01	2023-02-13	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	12	1	2
3	2003	Cloud Storage	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
4	2004	Cloud Application Hosting	2023-04-01	2023-04-13	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	12	1	2
5	2005	Cloud Security Solutions	2023-05-01	2023-05-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
6	2006	Private Cloud Solutions	2023-06-01	2023-06-14	4500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
7	2007	Cloud Platform Development	2023-07-01	2023-07-14	6000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
8	2008	Disaster Recovery Cloud	2023-08-01	2023-08-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
9	2009	Cloud Data Analytics	2023-09-01	2023-09-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
10	2010	Cloud DevOps Services	2023-10-01	2023-10-14	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
11	2011	Cloud Migration Solutions	2023-11-01	2023-11-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
12	2012	Cloud API Integration	2023-12-01	2023-12-14	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
13	2013	Hybrid Cloud Solutions	2023-01-15	2023-01-29	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	14	1	2
ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo	
1	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2

19. UpdateService

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[UpdateService]    Script Date: 1/4/2025 1:43:08 PM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[UpdateService]
    @ServiceID INT,
    @Name VARCHAR(50) = NULL,
    @StartDate DATE = NULL,
    @EndDate DATE = NULL,
    @Price INT = NULL,
    @Status VARCHAR(50) = NULL,
    @City VARCHAR(50) = NULL,
    @State VARCHAR(50) = NULL,
    @Country VARCHAR(50) = NULL,
    @Cloud VARCHAR(50) = NULL,
    @Security VARCHAR(50) = NULL,
    @Repair VARCHAR(50) = NULL,
    @Network VARCHAR(50) = NULL,
    @SoftwareDev VARCHAR(50) = NULL,
    @CompanyID INT = NULL,
    @DepartmentNo INT = NULL
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the specified ServiceID exists in the Service table.
        IF NOT EXISTS (SELECT 1 FROM Service WHERE ServiceID = @ServiceID)
        BEGIN
            THROW 53003, 'ServiceID not found. Update operation aborted.', 1;
        END;

        -- Perform the update operation, using existing values if no new values are provided.
        UPDATE Service
        SET
            Name = ISNULL(@Name, Name),
            StartDate = ISNULL(@StartDate, StartDate),
            EndDate = ISNULL(@EndDate, EndDate),
            Price = ISNULL(@Price, Price),
            Status = ISNULL(@Status, Status),
            City = ISNULL(@City, City),
            State = ISNULL(@State, State),
            Country = ISNULL(@Country, Country),
            Cloud = ISNULL(@Cloud, Cloud),
            Security = ISNULL(@Security, Security),
```

100 %

Connected. (1/1)

```

BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the specified ServiceID exists in the Service table.
        IF NOT EXISTS (SELECT 1 FROM Service WHERE ServiceID = @ServiceID)
        BEGIN
            THROW 53003, 'ServiceID not found. Update operation aborted.', 1;
        END;

        -- Perform the update operation, using existing values if no new values are provided.
        UPDATE Service
        SET
            Name = ISNULL(@Name, Name),
            StartDate = ISNULL(@StartDate, StartDate),
            EndDate = ISNULL(@EndDate, EndDate),
            Price = ISNULL(@Price, Price),
            Status = ISNULL(@Status, Status),
            City = ISNULL(@City, City),
            State = ISNULL(@State, State),
            Country = ISNULL(@Country, Country),
            Cloud = ISNULL(@Cloud, Cloud),
            Security = ISNULL(@Security, Security),
            Repair = ISNULL(@Repair, Repair),
            Network = ISNULL(@Network, Network),
            SoftwareDev = ISNULL(@SoftwareDev, SoftwareDev),
            CompanyID = ISNULL(@CompanyID, CompanyID),
            DepartmentNo = ISNULL(@DepartmentNo, DepartmentNo)
        WHERE ServiceID = @ServiceID;

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO

```

```
-- Update
SELECT * FROM Service;

EXEC InsertService
    @ServiceID = 1,
    @Name = 'Cloud Hosting',
    @StartDate = '2024-01-01',
    @EndDate = '2024-12-31'
```

100 %

Results Messages

	ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo
1	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
2	2002	Cloud Backup Services	2023-02-01	2023-02-13	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	12	1	2
3	2003	Cloud Storage	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
4	2004	Cloud Application Hosting	2023-04-01	2023-04-13	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	12	1	2
5	2005	Cloud Security Solutions	2023-05-01	2023-05-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
6	2006	Private Cloud Solutions	2023-06-01	2023-06-14	4500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
7	2007	Cloud Platform Development	2023-07-01	2023-07-14	6000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
8	2008	Disaster Recovery Cloud	2023-08-01	2023-08-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
9	2009	Cloud Data Analytics	2023-09-01	2023-09-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
10	2010	Cloud DevOps Services	2023-10-01	2023-10-14	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
11	2011	Cloud Migration Solutions	2023-11-01	2023-11-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
12	2012	Cloud API Integration	2023-12-01	2023-12-14	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
13	2013	Hybrid Cloud Solutions	2023-01-15	2023-01-29	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	14	1	2
14	2014	Cloud Infrastructure Management	2023-02-15	2023-02-28	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
15	2015	Multi-Cloud Solutions	2023-03-15	2023-03-28	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
16	2016	Cloud Computing Solutions	2023-04-15	2023-04-28	6000	Active	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
17	2017	Cloud Database Solutions	2023-05-15	2023-05-28	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
18	2018	Cloud Storage for Enterprises	2023-06-15	2023-06-28	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
19	2019	Cloud-Based Business Solutions	2023-07-15	2023-07-28	5000	Active	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
20	2020	Cloud Disaster Recovery	2023-08-15	2023-08-28	6000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
21	2021	Cloud SaaS Solutions	2023-09-15	2023-09-28	5500	Active	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
22	2022	Cloud Virtualization	2023-10-15	2023-10-28	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
23	2023	Cloud Data Storage	2023-11-15	2023-11-28	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
24	2024	Cloud Software Solutions	2023-12-15	2023-12-28	5500	Active	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
25	2025	Cloud Application Management	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
26	2026	Cloud Automation Services	2023-02-01	2023-02-14	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
27	2027	Cloud Hosting with Security	2023-03-01	2023-03-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
28	2028	Cloud Integration Solutions	2023-04-01	2023-04-14	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
29	2029	Cloud Security & Compliance	2023-05-01	2023-05-14	6000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
30	2030	Private Cloud Storage	2023-06-01	2023-06-14	5000	Active	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
31	3001	Network Firewall	2023-01-01	2023-01-15	4000	Completed	Istanbul	Marmara	Turkey	nan	Security	nan	nan	nan	14	1	6
32	3002	Antivirus Protection	2023-02-01	2023-02-14	1500	Completed	Izmir	Izmir	Turkey	nan	Security	nan	nan	nan	13	1	6
33	3003	Security Monitoring	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	nan	Security	nan	nan	nan	13	1	6
34	3004	Endpoint Protection	2023-04-01	2023-04-14	2500	Completed	Istanbul	Marmara	Turkey	nan	Security	nan	nan	nan	13	1	6
35	3005	Cloud Security	2023-05-01	2023-05-14	4000	Completed	Izmir	Izmir	Turkey	nan	Security	nan	nan	nan	13	1	6

Query executed successfully.

localhost\MSSQLSERVER01 (1)

```
EXEC UpdateService
    @ServiceID = 1,
    @Name = 'Advanced Cloud Hosting',
    @Price = 6000,
    @Cloud = 'Cloud';

EXEC UpdateService
    @ServiceID = 2,
    @Name = 'Advanced Cloud Hosting',
    @Price = 6000,
    @Status = 'Completed',
    @Security = 'security';

SELECT * FROM Service;
```

100 %

Results Messages

	ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo
1	1	Advanced Cloud Hosting	2024-01-01	2024-12-31	6000	Active	New York	NY	USA	Cloud	NULL	NULL	NULL	NULL	365	1	101
2	2	Advanced Cloud Hosting	2024-01-01	2024-12-31	6000	Completed	San Francisco	CA	USA	NULL	security	NULL	NULL	NULL	365	1	101
3	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
4	2002	Cloud Backup Services	2023-02-01	2023-02-13	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	12	1	2
5	2003	Cloud Storage	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
6	2004	Cloud Application Hosting	2023-04-01	2023-04-13	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	12	1	2
7	2005	Cloud Security Solutions	2023-05-01	2023-05-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
8	2006	Private Cloud Solutions	2023-06-01	2023-06-14	4500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
9	2007	Cloud Platform Development	2023-07-01	2023-07-14	6000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
10	2008	Disaster Recovery Cloud	2023-08-01	2023-08-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
11	2009	Cloud Data Analytics	2023-09-01	2023-09-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
12	2010	Cloud DevOps Services	2023-10-01	2023-10-14	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
13	2011	Cloud Migration Solutions	2023-11-01	2023-11-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
14	2012	Cloud API Integration	2023-12-01	2023-12-14	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
15	2013	Hybrid Cloud Solutions	2023-01-15	2023-01-29	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	14	1	2
16	2014	Cloud Infrastructure Mana...	2023-02-15	2023-02-28	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
17	2015	Multi-Cloud Solutions	2023-03-15	2023-03-28	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
18	2016	Cloud Computing Solutions	2023-04-15	2023-04-28	6000	Active	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
19	2017	Cloud Database Solutions	2023-05-15	2023-05-28	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2

Query executed successfully.

localhost\MSSQLSERVER01

20. DeleteService

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

```
USE [Database-001]
GO

/***** Object: StoredProcedure [dbo].[DeleteService]    Script Date: 1/4/2025 1:29:29 PM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-----

CREATE PROCEDURE [dbo].[DeleteService]
    @ServiceID INT
AS
BEGIN
    -- Start a transaction to ensure atomicity of the operation.
    BEGIN TRANSACTION;

    BEGIN TRY
        -- Check if the specified ServiceID exists in the Service table.
        IF NOT EXISTS (SELECT 1 FROM Service WHERE ServiceID = @ServiceID)
        BEGIN
            THROW 53004, 'ServiceID not found. Delete operation aborted.', 1;
        END;

        -- Delete related records in ServiceLog.
        DELETE FROM ServiceLog WHERE ServiceID = @ServiceID;

        -- Perform the delete operation on the Service table.
        DELETE FROM Service WHERE ServiceID = @ServiceID;

        -- Commit the transaction if no errors occurred.
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        -- Rollback the transaction in case of an error.
        ROLLBACK TRANSACTION;
        -- Re-throw the error for higher-level handling.
        THROW;
    END CATCH;
END;
GO
```

-- Delete

```
SELECT * FROM Service;  
SELECT * FROM ServiceLog;
```

```
EXEC DeleteService @ServiceID = 1;
```

```
SELECT * FROM Service;  
SELECT * FROM ServiceLog;
```

00 %

Results Messages

	ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo
1	1	Cloud Hosting	2024-01-01	2024-12-31	5000	Active	New York	NY	USA	AWS	NULL	NULL	NULL	NULL	365	1	101
2	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
3	2002	Cloud Backup Services	2023-02-01	2023-02-13	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	12	1	2
4	2003	Cloud Storage	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
5	2004	Cloud Application Hosting	2023-04-01	2023-04-13	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	12	1	2
6	2005	Cloud Security Solutions	2023-05-01	2023-05-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
7	2006	Private Cloud Solutions	2023-06-01	2023-06-14	4500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
8	2007	Cloud Platform Development	2023-07-01	2023-07-14	6000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
9	2008	Disaster Recovery Cloud	2023-08-01	2023-08-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
10	2009	Cloud Data Analytics	2023-09-01	2023-09-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
11	2010	Cloud DevOps Services	2023-10-01	2023-10-14	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
12	2011	Cloud Migration Solutions	2023-11-01	2023-11-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2

	LogID	ServiceID	Action	LogDate
1	5	1	INSERT	2025-01-04 13:24:35.037

-- Delete

```
SELECT * FROM Service;  
SELECT * FROM ServiceLog;
```

```
EXEC DeleteService @ServiceID = 1;
```

```
SELECT * FROM Service;  
SELECT * FROM ServiceLog;
```

100 %

Results Messages

	ServiceID	Name	StartDate	EndDate	Price	Status	City	State	Country	Cloud	Security	Repair	Network	SoftwareDev	Duration	CompanyID	DepartmentNo
1	2001	Cloud Hosting	2023-01-01	2023-01-14	5000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
2	2002	Cloud Backup Services	2023-02-01	2023-02-13	4500	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	12	1	2
3	2003	Cloud Storage	2023-03-01	2023-03-14	3500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
4	2004	Cloud Application Hosting	2023-04-01	2023-04-13	4000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	12	1	2
5	2005	Cloud Security Solutions	2023-05-01	2023-05-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
6	2006	Private Cloud Solutions	2023-06-01	2023-06-14	4500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
7	2007	Cloud Platform Development	2023-07-01	2023-07-14	6000	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
8	2008	Disaster Recovery Cloud	2023-08-01	2023-08-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
9	2009	Cloud Data Analytics	2023-09-01	2023-09-14	5500	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
10	2010	Cloud DevOps Services	2023-10-01	2023-10-14	4500	Completed	Istanbul	Marmara	Turkey	Cloud	nan	nan	nan	nan	13	1	2
11	2011	Cloud Migration Solutions	2023-11-01	2023-11-14	5000	Completed	Izmir	Izmir	Turkey	Cloud	nan	nan	nan	nan	13	1	2
12	2012	Cloud API Integration	2023-12-01	2023-12-14	4000	Completed	Ankara	Ankara	Turkey	Cloud	nan	nan	nan	nan	13	1	2

	LogID	ServiceID	Action	LogDate
--	-------	-----------	--------	---------

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**, **Clustered indices** 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.
