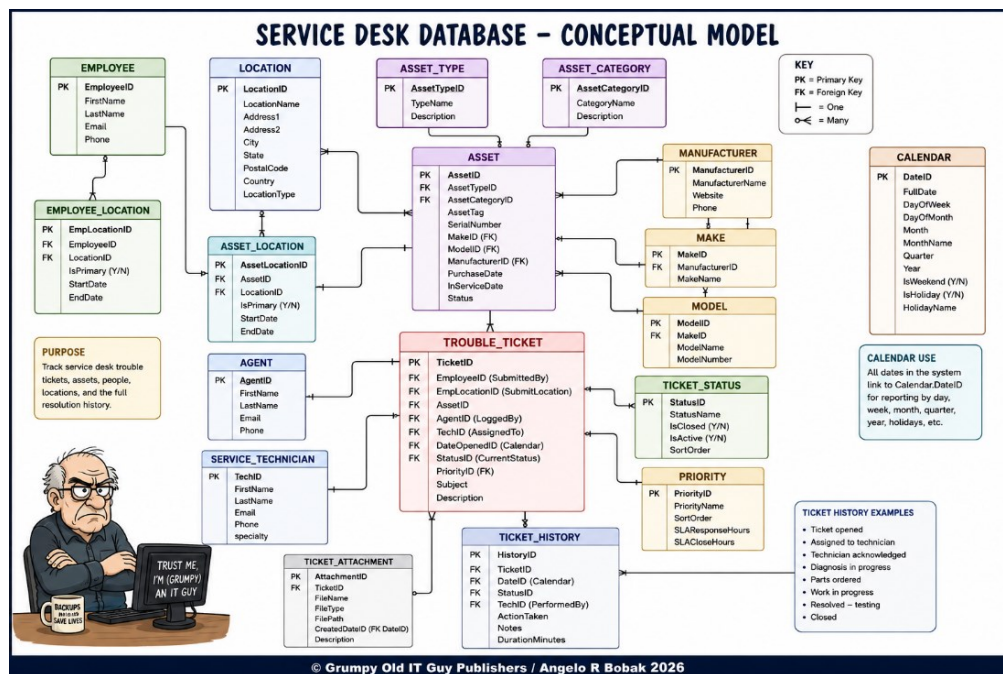


Service Desk – Conceptual Model

Grumpy Old IT Guy Publishers and **Angelo R. Bobak** are independent educational content providers and are not affiliated with, endorsed by, sponsored by, or associated with Microsoft Corporation.

All references to Microsoft technologies are used strictly for educational, instructional, commentary, and demonstration purposes under the principles of nominative fair use.

Any screenshots, product references, feature descriptions, or code examples referencing Microsoft technologies are intended solely to help users learn database design, SQL development, and related technical concepts.



A Service Desk system serves as the operational nerve center for managing technology assets, employee support requests, and the lifecycle of technical incidents within an organization. In most modern companies, employees depend heavily on computers, mobile devices, printers, network infrastructure, servers, and software applications to perform daily business activities. When a device fails, software malfunctions, network connectivity is interrupted, or hardware performance degrades, the Service Desk becomes responsible for recording the issue, assigning technical resources, tracking diagnostic and repair activities, and ultimately restoring service to normal operations.

At the core of the Service Desk business concept is accountability and traceability. Every trouble ticket is a documented business event that captures who reported the issue, which asset was affected, where the issue occurred, when it was reported, and which support personnel were involved in resolving the problem. This historical information becomes valuable not only for operational support but also for management reporting, capacity planning, budgeting, vendor analysis, compliance auditing, and long-term technology strategy. By maintaining detailed ticket histories, organizations can identify recurring hardware failures, problematic software deployments, overloaded support teams, or locations experiencing persistent technical difficulties.

A well-designed Service Desk database also supports enterprise asset management. Assets such as laptops, servers, printers, storage devices, and network equipment are tracked throughout their lifecycle, including procurement, deployment, relocation, maintenance, and retirement. Each asset can be associated with manufacturers, models, serial numbers, locations, assigned employees, and support histories. This allows organizations to manage inventory more effectively, control technology costs, improve security tracking, and maintain accurate records for audits and insurance purposes.

Another important business function of the Service Desk is workflow coordination. Trouble tickets move through a sequence of statuses such as Open, Assigned, In Progress, Waiting for Parts, Resolved, and Closed. During this lifecycle, support agents, service technicians, managers, and sometimes outside vendors collaborate to diagnose and resolve issues. The database preserves each step of this process, creating a complete operational timeline that can be analyzed later to measure service quality, technician productivity, and compliance with Service Level Agreements (SLAs).

The inclusion of a Calendar dimension within the database further enhances analytical capabilities. Calendar-based reporting allows management to analyze ticket volumes by day, week, month, quarter, or year; identify seasonal trends; monitor holiday support activity; and measure response and resolution times. This type of reporting is especially valuable for IT management teams responsible for staffing decisions, budgeting, hardware replacement planning, and operational performance monitoring.

From a broader business perspective, the Service Desk functions as both a customer support platform and a corporate knowledge repository. Over time, ticket histories accumulate valuable technical knowledge regarding common failures, repair procedures, vendor reliability, recurring user issues, and infrastructure weaknesses. This information can be used to improve support processes, standardize troubleshooting procedures, train technicians, and reduce future downtime across the organization.

In large enterprises, the Service Desk often integrates with related operational disciplines such as asset management, configuration management, cybersecurity monitoring, software deployment, procurement systems, and data governance frameworks. Together, these systems help organizations maintain stable technology operations while providing measurable visibility into the health, cost, and effectiveness of the company's technical infrastructure.

This Service Desk and Asset Management database model, therefore, provides the foundation for managing technical operations in a structured, auditable, and scalable manner. By combining ticket management, employee support, asset tracking, technician workflow, historical event recording, and calendar-based analytics, the database supports both day-to-day operational activities and long-term strategic decision-making within the enterprise IT environment.

Entity Data Dictionary

Below is the data dictionary describing the entities of the conceptual data model.

Entity Name	Description / Purpose
Location	Stores physical business locations such as headquarters, branch offices, warehouses, data centers, and remote offices where employees or assets are located.
Employee	Stores information about employees who submit trouble tickets or are associated with company assets and support activities.
EmployeeLocation	Associates employees with one or more physical business locations and tracks their assignments over time.
AssetType	Defines the broad classification of an asset, such as a desktop computer, a laptop, a server, a printer, or a network device.
AssetCategory	Defines higher-level groupings of assets used for reporting, inventory management, budgeting, and operational analysis.
Manufacturer	Stores information about companies that manufacture technology assets, such as Dell, Cisco, Lenovo, or HP.
Make	Stores the commercial brand or product line associated with a manufacturer.
Model	Stores detailed model information for assets, including model names and model numbers used for support and inventory tracking.
Asset	Stores the master inventory records for all company-owned technology assets, including asset tags, serial numbers, hardware classifications, and operational status.
AssetLocation	Tracks the physical location history of assets as they move between offices, departments, warehouses, or data centers.
Agent	Stores information about service desk agents who log and manage incoming trouble tickets.
ServiceTechnician	Stores information about technical support personnel responsible for diagnosing, repairing, and resolving technical issues associated with trouble tickets.
Calendar	Provides a reusable calendar dimension used for reporting, analytics, SLA calculations, trend analysis, and time-based historical tracking.
TicketStatus	Defines the lifecycle states of a trouble ticket, such as Open, Assigned, In Progress, Resolved, and Closed.

Entity Name	Description / Purpose
Priority	Defines the business severity and urgency levels associated with trouble tickets, including SLA response and resolution expectations.
TroubleTicket	Stores the master records for technical incidents, service requests, and support events reported by employees regarding company assets or services.
TicketHistory	Stores the detailed chronological history of actions, status changes, technician activities, diagnostic steps, and resolution notes associated with a trouble ticket.
TicketAttachment	Stores metadata describing files, screenshots, logs, documents, or other electronic attachments associated with trouble tickets.

Entity Relationship Data Dictionary

Below is the data dictionary describing the relationships among the entities.

Parent Entity	Relationship	Child Entity
Location	contains assigned employees through	EmployeeLocation
Employee	is assigned to	EmployeeLocation
Location	contains deployed assets through	AssetLocation
Asset	is assigned to	AssetLocation
Manufacturer	produces	Make
Make	defines product lines for	Model
Manufacturer	manufactures	Asset
Make	identifies the commercial brand of	Asset
Model	identifies the hardware model of	Asset
AssetType	classifies	Asset
AssetCategory	categorizes	Asset
Employee	submits	TroubleTicket
EmployeeLocation	identifies the reporting location for	TroubleTicket
Asset	is associated with	TroubleTicket
Agent	logs and manages	TroubleTicket
ServiceTechnician	is assigned to resolve	TroubleTicket
TicketStatus	defines the current status of	TroubleTicket
Priority	defines the urgency level of	TroubleTicket
Calendar	records the opening date for	TroubleTicket
Calendar	records the closing date for	TroubleTicket
TroubleTicket	contains historical activity recorded in	TicketHistory

Parent Entity	Relationship	Child Entity
TicketStatus	identifies the status recorded in	TicketHistory
ServiceTechnician	performs work activities recorded in	TicketHistory
Calendar	records the activity date for	TicketHistory
TroubleTicket	contains supporting documents in	TicketAttachment
Calendar	records the creation date of	TicketAttachment
Location	supports operational activities related to	TroubleTicket
Asset	generates operational support history through	TicketHistory
ServiceTechnician	updates resolution activities within	TicketHistory
Agent	initiates support workflow for	TroubleTicket
Priority	establishes SLA expectations for	TroubleTicket
Calendar	supports trend and historical reporting for	TroubleTicket
Calendar	supports time-based analytics for	TicketHistory

Disclaimer

The conceptual models, logical models, database structures, Microsoft SQL Server scripts, analytical queries, diagrams, documentation, and supporting materials contained in this publication were developed exclusively as instructional examples intended for learning, experimentation, evaluation, and preliminary prototype development.

These materials are not represented as certified production systems, enterprise-ready architectures, or operational standards. Readers and users are solely responsible for independently reviewing, testing, validating, securing, and adapting all database objects, code samples, and architectural designs before implementation in any live or business-critical environment.

Grumpy Old IT Guy Publishers and Angelo R Bobak make no guarantees regarding performance, accuracy, compatibility, security, reliability, regulatory compliance, or operational suitability. All examples and technical content are supplied without warranty of any kind, either expressed or implied.

By using any portion of these materials, the reader acknowledges that all implementation decisions, modifications, deployments, and operational usage are performed entirely at the user's own discretion and risk. Grumpy Old IT Guy Publishers and Angelo R Bobak shall not be responsible for any losses, damage, outages, liabilities, compliance issues, data corruption, or business interruptions that may result from the use or misuse of these materials.

All examples are intended strictly for educational and demonstration purposes.

© Grumpy Old IT Guy Publishers / Angelo R Bobak 2026

