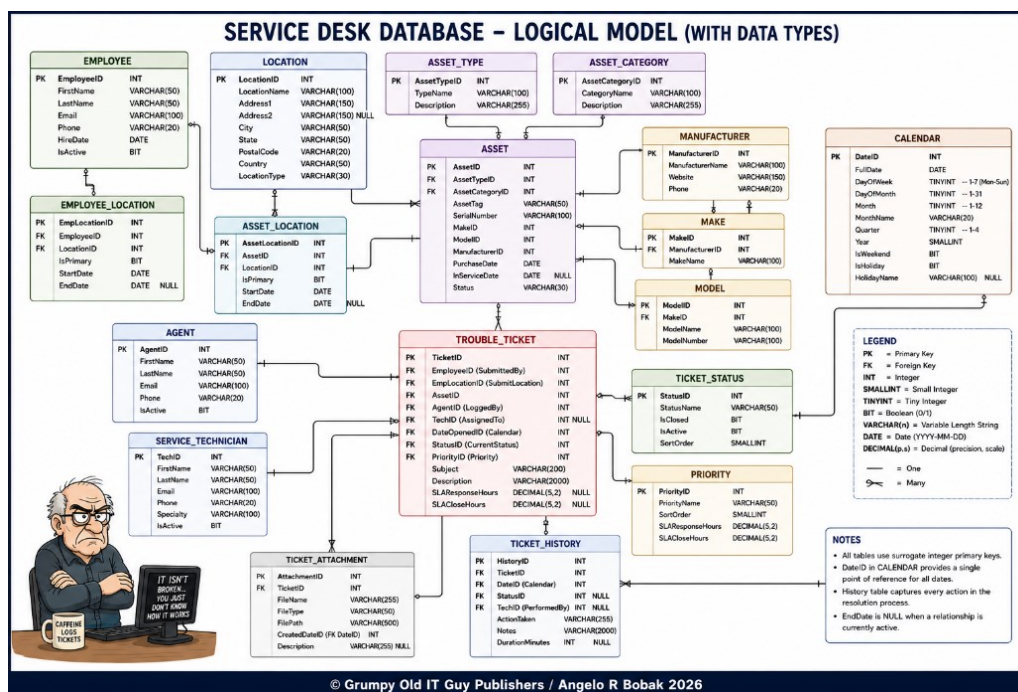


Service Desk – Logical Model

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.



The logical Service Desk and Asset Management data model provides a structured representation of how technical support operations, company assets, employees, and support activities interact within an enterprise environment. The purpose of the model is to organize operational support data into well-defined entities that can efficiently support ticket management, asset tracking, workflow processing, historical auditing, and analytical reporting.

At the center of the logical model is the TroubleTicket entity, which represents a reported technical issue, service request, or operational incident. Each ticket captures information about the employee who submitted the issue, the asset associated with the problem, the service desk agent who logged the ticket, the assigned service technician responsible for resolving the issue,

the ticket priority, and the current ticket status. This entity serves as the primary operational transaction table in the database, recording the ongoing support activities performed by the organization.

The model also includes a detailed Asset Management structure. Assets are organized into several supporting entities including AssetType, AssetCategory, Manufacturer, Make, and Model. This layered structure allows the organization to standardize inventory classification while supporting analytical reporting across hardware families, vendors, device categories, and operational asset groups. The Asset entity itself stores the individual inventory records for physical equipment such as desktop computers, laptops, servers, printers, storage systems, and networking devices.

Because assets and employees may move between offices or facilities over time, the model includes both AssetLocation and EmployeeLocation entities. These tables provide historical tracking capabilities that allow the organization to determine where an employee or asset was located at a specific point in time. This becomes particularly important for operational auditing, security reviews, inventory control, and support trend analysis across multiple business locations.

The logical model also separates operational support personnel into distinct functional entities. The Agent entity represents service desk personnel responsible for receiving and logging trouble calls, while the ServiceTechnician entity represents the technical specialists responsible for diagnosing and resolving the reported issues. This separation supports workload analysis, staffing metrics, productivity reporting, and escalation management.

An important feature of the model is the inclusion of the TicketHistory entity, which records the chronological lifecycle of each support ticket. Every assignment, diagnostic step, status change, technician update, escalation, repair action, and resolution activity can be recorded within this entity. By maintaining this detailed history, the organization gains full operational traceability and can analyze support performance, resolution trends, recurring failures, and SLA compliance over time.

The Calendar entity provides a reusable date dimension that supports enterprise reporting and analytics. Rather than storing only raw date values, the calendar structure allows support data to be analyzed by day, week, month, quarter, year, fiscal period, business day, weekend, or holiday. This design improves reporting consistency while simplifying analytical queries and dashboard generation.

Additional operational support is provided through entities such as TicketStatus, which defines the workflow stages of support tickets, Priority, which defines business urgency and SLA expectations, and TicketAttachment, which stores metadata related to screenshots, log files, diagnostic reports, or supporting documentation attached to support cases.

Overall, the logical model was designed to support the complete lifecycle of enterprise support management, including asset inventory tracking, issue reporting, technician workflow coordination, operational auditing, service analytics, SLA monitoring, and historical reporting. The design follows normalized relational modeling principles intended to reduce data redundancy, improve maintainability, and provide a scalable foundation for future application development, reporting systems, dashboards, and business intelligence solutions.

Entity/Attribute Data Dictionary

This data dictionary identifies the attributes of each entity, along with their Microsoft SQL Server data types.

Entity	Attribute	Data Type
Location	LocationID	INT
Location	LocationName	VARCHAR(100)
Location	Address1	VARCHAR(150)
Location	Address2	VARCHAR(150)
Location	City	VARCHAR(50)
Location	StateProvince	VARCHAR(50)
Location	PostalCode	VARCHAR(20)
Location	Country	VARCHAR(50)
Location	LocationType	VARCHAR(30)
Location	IsActive	BIT
Employee	EmployeeID	INT
Employee	FirstName	VARCHAR(50)
Employee	LastName	VARCHAR(50)
Employee	Email	VARCHAR(100)
Employee	Phone	VARCHAR(20)
Employee	HireDate	DATE
Employee	IsActive	BIT
EmployeeLocation	EmployeeLocationID	INT
EmployeeLocation	EmployeeID	INT
EmployeeLocation	LocationID	INT
EmployeeLocation	IsPrimary	BIT
EmployeeLocation	StartDate	DATE
EmployeeLocation	EndDate	DATE
AssetType	AssetTypeID	INT
AssetType	TypeName	VARCHAR(100)
AssetType	Description	VARCHAR(255)
AssetCategory	AssetCategoryID	INT

Entity	Attribute	Data Type
AssetCategory	CategoryName	VARCHAR(100)
AssetCategory	Description	VARCHAR(255)
Manufacturer	ManufacturerID	INT
Manufacturer	ManufacturerName	VARCHAR(100)
Manufacturer	Website	VARCHAR(150)
Manufacturer	Phone	VARCHAR(20)
Make	MakeID	INT
Make	ManufacturerID	INT
Make	MakeName	VARCHAR(100)
Model	ModelID	INT
Model	MakeID	INT
Model	ModelName	VARCHAR(100)
Model	ModelNumber	VARCHAR(100)
Asset	AssetID	INT
Asset	AssetTypeID	INT
Asset	AssetCategoryID	INT
Asset	AssetTag	VARCHAR(50)
Asset	SerialNumber	VARCHAR(100)
Asset	MakeID	INT
Asset	ModelID	INT
Asset	ManufacturerID	INT
Asset	PurchaseDate	DATE
Asset	InServiceDate	DATE
Asset	AssetStatus	VARCHAR(30)
AssetLocation	AssetLocationID	INT
AssetLocation	AssetID	INT
AssetLocation	LocationID	INT
AssetLocation	IsPrimary	BIT
AssetLocation	StartDate	DATE
AssetLocation	EndDate	DATE
Agent	AgentID	INT
Agent	FirstName	VARCHAR(50)
Agent	LastName	VARCHAR(50)
Agent	Email	VARCHAR(100)
Agent	Phone	VARCHAR(20)
Agent	IsActive	BIT
ServiceTechnician	TechID	INT

Entity	Attribute	Data Type
ServiceTechnician	FirstName	VARCHAR(50)
ServiceTechnician	LastName	VARCHAR(50)
ServiceTechnician	Email	VARCHAR(100)
ServiceTechnician	Phone	VARCHAR(20)
ServiceTechnician	Specialty	VARCHAR(100)
ServiceTechnician	IsActive	BIT
Calendar	DateID	INT
Calendar	FullDate	DATE
Calendar	DayOfWeek	TINYINT
Calendar	DayOfMonth	TINYINT
Calendar	MonthNumber	TINYINT
Calendar	MonthName	VARCHAR(20)
Calendar	QuarterNumber	TINYINT
Calendar	YearNumber	SMALLINT
Calendar	IsWeekend	BIT
Calendar	IsHoliday	BIT
Calendar	HolidayName	VARCHAR(100)
TicketStatus	StatusID	INT
TicketStatus	StatusName	VARCHAR(50)
TicketStatus	IsClosed	BIT
TicketStatus	IsActive	BIT
TicketStatus	SortOrder	SMALLINT
Priority	PriorityID	INT
Priority	PriorityName	VARCHAR(50)
Priority	SortOrder	SMALLINT
Priority	SLAResponseHours	DECIMAL(5,2)
Priority	SLACloseHours	DECIMAL(5,2)
TroubleTicket	TicketID	INT
TroubleTicket	EmployeeID	INT
TroubleTicket	EmpLocationID	INT
TroubleTicket	AssetID	INT
TroubleTicket	AgentID	INT
TroubleTicket	TechID	INT
TroubleTicket	DateOpenedID	INT
TroubleTicket	DateClosedID	INT
TroubleTicket	StatusID	INT
TroubleTicket	PriorityID	INT

Entity	Attribute	Data Type
TroubleTicket	Subject	VARCHAR(200)
TroubleTicket	Description	VARCHAR(2000)
TroubleTicket	SLAResponseHours	DECIMAL(5,2)
TroubleTicket	SLACloseHours	DECIMAL(5,2)
TicketHistory	HistoryID	INT
TicketHistory	TicketID	INT
TicketHistory	DateID	INT
TicketHistory	StatusID	INT
TicketHistory	TechID	INT
TicketHistory	ActionTaken	VARCHAR(255)
TicketHistory	Notes	VARCHAR(2000)
TicketHistory	DurationMinutes	INT
TicketAttachment	AttachmentID	INT
TicketAttachment	TicketID	INT
TicketAttachment	FileName	VARCHAR(255)
TicketAttachment	FileType	VARCHAR(50)
TicketAttachment	FilePath	VARCHAR(500)
TicketAttachment	CreatedDateID	INT
TicketAttachment	Description	VARCHAR(255)

Entity/Relationship Data Dictionary

This data dictionary identifies the primary and foreign keys for entities participating in a relationship:

Parent Entity	Parent Primary Key	Relationship	Child Entity	Child Foreign Key
Location	LocationID	identifies the assigned employee location for	EmployeeLocation	LocationID
Employee	EmployeeID	is assigned to one or more locations through	EmployeeLocation	EmployeeID
Location	LocationID	identifies the deployed asset location for	AssetLocation	LocationID

Parent Entity	Parent Primary Key	Relationship	Child Entity	Child Foreign Key
Asset	AssetID	is assigned to one or more locations through	AssetLocation	AssetID
Manufacturer	ManufacturerID	produces one or more makes	Make	ManufacturerID
Make	MakeID	defines one or more models	Model	MakeID
Manufacturer	ManufacturerID	identifies the manufacturer of	Asset	ManufacturerID
Make	MakeID	identifies the make of	Asset	MakeID
Model	ModelID	identifies the model of	Asset	ModelID
AssetType	AssetTypeID	classifies	Asset	AssetTypeID
AssetCategory	AssetCategoryID	categorizes	Asset	AssetCategoryID
Employee	EmployeeID	submits	TroubleTicket	EmployeeID
EmployeeLocation	EmployeeLocationID	identifies the employee reporting location for	TroubleTicket	EmpLocationID
Asset	AssetID	is the asset involved in	TroubleTicket	AssetID
Agent	AgentID	logs and manages	TroubleTicket	AgentID
ServiceTechnician	TechID	is assigned to resolve	TroubleTicket	TechID
TicketStatus	StatusID	defines the current workflow status of	TroubleTicket	StatusID
Priority	PriorityID	defines the urgency and SLA level of	TroubleTicket	PriorityID
Calendar	DateID	records the ticket opening date for	TroubleTicket	DateOpenedID
Calendar	DateID	records the ticket closing date for	TroubleTicket	DateClosedID

Parent Entity	Parent Primary Key	Relationship	Child Entity	Child Foreign Key
TroubleTicket	TicketID	contains one or more historical activity records in	TicketHistory	TicketID
Calendar	DateID	records the activity date for	TicketHistory	DateID
TicketStatus	StatusID	identifies the status recorded for	TicketHistory	StatusID
ServiceTechnician	TechID	identifies the technician who performed work recorded in	TicketHistory	TechID
TroubleTicket	TicketID	contains one or more supporting attachments in	TicketAttachment	TicketID
Calendar	DateID	records the attachment creation date for	TicketAttachment	CreatedDateID

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