

Assignment 1 – Speeding Tickets Database Design

Purpose

To design an effective helpdesk ticket system, nicknamed “Speeding Tickets”, that will be used in a standard IT department to manage issues.

Background

Every IT department can benefit from a helpdesk ticket system. It allows users to submit issues to the IT department with ease. It keeps the IT department organized and efficient, keeping a record of problems and solutions while keeping stress down. It allows a continuous channel of communication between the users and the technicians, and it ensures that everyone is served in a timely and ordered manner.

Requirements

1. The database must hold information about the users and the support staff.
2. Both users and support staff can generate new tickets.
3. Tickets must at least contain information about who generated the ticket, what category the ticket is under (e.g. software installation, hardware issue, etc), and who is currently assigned to work on the ticket.
4. If the ticket is for a hardware issue, the ticket will also contain information about the hardware, so any warranty repairs can be dealt with.
5. Once the ticket has been submitted, a support staff member will then be assigned to handle the ticket.
6. The staff member may request additional information from the user for clarification around the issue and this will be communicated via a comment thread in the ticket that the user can then respond to.
7. The ticket may also generate different tasks that need to be tracked as well. For example, new equipment may need to be ordered, installed or configured.

Task

1. Create an Entity-Relationship Diagram for your ticket database using Visual Paradigm.
 - a. The diagram must use Crow’s Foot Notation.
 - b. The diagram must use appropriate data types and field lengths.
 - c. The diagram must use SQL Server 2019 as the target database.
2. Create a Data Dictionary document that contains enough information to completely document your database design.
 - a. The document must describe the purpose of each table.
 - b. For each field, the document must describe the purpose of the field, the size and type of the field and **any** constraints on the field (e.g. PK, FK, etc.).
 - c. Include enough sample data for each table and field to demonstrate proper use.
 - d. The document must be created as a version-controlled, Markdown or AsciiDoc document in GitHub.

RUBRIC

	Criteria	Unsatisfactory	Acceptable	Good	Exceptional	Mark
		0	1	2	3	
Entity Relationship Diagram	Design	- database does not satisfy the requirements	- the database design will suit most of the requirements, but more work is needed	- the database is a good design and will make for a usable bug tracking system	- an excellent database design that shows foresight, flexibility and maintainability	<u> </u> x3
	Entities	- not in 3NF - missing entities	- in 3NF - some primary keys defined - a few tables are poorly chosen or could be combined	- in 3NF - most primary keys defined correctly - only a few errors exist - could have chosen better names	- in 3NF - all primary keys are correctly defined - all entities are well chosen and appropriately named	<u> </u>
	Relationships	- no relationships defined - no foreign keys visible	- defined, but some directions are reversed - not using Crow's Foot Notation	- defined correctly, but foreign keys don't match to primary keys	- relationships and foreign keys defined correctly - using Crow's Foot Notation	<u> </u>
	Data Types and Data Lengths	- most data types/lengths are left to default - e.g. CHAR(10) - data types are not shown on the ERD	- most data types defined correctly - some errors exist - did not select SQL Server Driver for all data types	- most data types are defined correctly - only a few errors exist - some data lengths are unsuitable	- all data types are defined correctly and all data lengths are appropriately determined	<u> </u>
	Fields	- not enough thought on necessary fields - most critical information missing	- some of the critical information missing - some errors exist	- most of the critical information exists as fields - a few errors exist	- all necessary information can be stored in given fields - no errors exist - all fields are well named	<u> </u>
Data Dictionary	Content	- dictionary does not exist - too many tables or fields are missing	- most tables are described - some fields are missing - other errors exist	- all tables/fields are described - only a few errors exist	- all tables/fields are included and are accurately described	<u> </u>
	Detail	- not enough information exists to accurately describe the database	- some fields and tables are well defined, but a lot more detail is required for clarity	- most fields and tables are well defined - a little more detail is required	- all tables and fields include all information necessary to completely understand the database	<u> </u>
					Total	<u> </u> 27