**Practical Database Design**

Supporting Documentation

V1.0.1

Table of Contents

[1. Introduction 3](#_Toc124018049)

[1.1. Course Goal 3](#_Toc124018050)

[1.2. Scope, Approach and Methods 3](#_Toc124018051)

[2. Blog Functional Support 3](#_Toc124018052)

[3. Functional Requirements 4](#_Toc124018053)

[Figure 1 Database functional requirements 4](#_Toc124018648)

# Introduction

Are Blogs even still relevant? Statistics suggest that personal blogs or blogging can be finically rewarding in the right niche. Regardless of the popularity of blogs, many, if not all, of the top 100 companies/brands have a blog presence. This popularity can be attributed to the fact that blogs are still considered a fundamental component of successful web marketing. A blog as a service provides information and interaction with new, existing, and potential customers and serves employees and other stakeholders.

## Course Goal

Our goal is to formulate a functional database design, also known as a database schema, to support a Blog application development for use by a business/brand entity or implemented as a personal Blog. It is important to remember that there is no single right or wrong way to design a database. The final database design can take many forms based on the complexity of the requirements.

## Scope, Approach and Methods

The database design will be developed for non-specific relational database technology. We assume that the initial stages of the project have taken place and that a complete requirements analysis has been undertaken for the blog application. We start our design process, having been provided with an extracted summary of functional support requirements. Therefore, we use the functional support requirements to shape the database design for a Blog application which will be derived by formulating and mapping entities as tables and attributes as table fields, assigning unique identifiers as unique keys and establishing relationships between entities. During the design process, we refine the tables, fields, and table relationships to support the requirements of a Blog application.

# Blog Functional Support

Functional requirements need to be clear, simple, and unambiguous. Figure 1 is a table of functional support requirements details, a list of features, behaviour, and requirements for a blog application that our database design must help facilitate.

|  |  |
| --- | --- |
| ID | Details |
| 1 | Easy retrieval of information, both text and image artefacts, related to blog posts. |
| 2 | Support the application to ensure that CRUD, create, return, update and delete operations can be performed. |
| 3 | Secure access to Blog data administration. |
| 4 | Blog authors to be able to upload data (images). |
| 5 | Support administration of data through a web-based interface. |
| 6 | Support querying / returning Blog entries-based date created. |
| 7 | Facilitate the option to assign status attributes to blog entries (Editing, Rework, Review, Published, Unpublished). |
| 8 | Support multiple user types (Admin, Editor, Creator). |
| 9 | Support querying / returning Blog entries based on category association |
| 10 | Ability to audit activities related to actions performed |
| 11 | Reader alerts to any objectionable material by flagging |
| 11 | Provide a flexible approach for associating custom attributes/meta-data to blog entries |
| ~~12~~ | ~~Be able to administer user access and permissions~~ |
| ~~13~~ | ~~Ability to audit activities related to actions performed on the database~~ |
| ~~14~~ | ~~Manage access to Blog data for different user types~~ |
| 15 | Support public commenting on blog posts |
| 16 | Support public rating of blog posts |