On Media Collaborative Database Design: A Postprint

Authors: Song Lingyu, Zhang Shuyong, Jingsen Kan

Date: 2023-10-08T00:00:00+00:00

Abstract

Databases serve as the core and foundation of information systems. Beyond facilitating convenient, timely, and accurate retrieval of required information, a well-designed database must also meet requirements for maintainability and scalability, while considering data consistency, redundancy, and access efficiency.

Full Text

A Brief Discussion on Media Collaboration Database Design

Abstract: As the core foundation of information systems, a well-designed database not only enables convenient, timely, and accurate information retrieval but also satisfies requirements for easy maintenance and expansion while considering data consistency, redundancy, and access efficiency.

Keywords: database; table; retrieval; index

3. Database Design (Package 07)

Since March 2015, as the design and development of the National Digital Composite Publishing System Engineering—Media Collaboration System (Package 07) has gradually unfolded, database design has become particularly critical. It determines whether the various functions of Package 07 can be tightly integrated and how they combine, making it a crucial component of Package 07's development and construction. This paper analyzes problems encountered in previous application system development to avoid these pitfalls and proposes a standardized, high-performance, maintainable, and extensible database design to meet Package 07's functional requirements and future scalability.

1. Problems in Previous Application Systems Through involvement in the design and development of several application systems, the authors have experienced bottlenecks and issues during project development, which manifest in several aspects. First, chaotic naming of data tables and fields: some projects feature non-standard naming conventions, making it difficult to locate required tables and causing significant inconvenience and confusion for developers. Second, difficulty in functional extension: some projects' data table fields barely meet initial functional requirements, necessitating the addition of fields or even entire tables during development when functionality needs to be extended. This can lead to program modifications or even redevelopment, increasing developer workload and potentially causing project delays. When such projects are delivered to multiple users, multi-version maintenance becomes necessary, increasing maintenance burdens. Third, slow data retrieval: although some projects were well-designed from the outset, avoiding many database design problems, as user data grows—especially reaching millions of records—data retrieval speed significantly decreases, causing inconvenience and frustration for users.

3.1 Naming Conventions The database design for Package 07 is standardized and optimized from several perspectives, beginning with naming conventions. **Tables:** T + representative letter + table name. Table names should consist of words or abbreviations with specific meanings that describe the table's content. For example: TISTORYSOURCE for manuscript source table. **Fields:** type prefix + field name, composed of characteristic words or abbreviations. **Primary keys:** PK_ + table name + constituent field name. If a composite primary key has many constituent fields, only the first field is included; the table name prefix may be omitted. **Foreign keys:** FK_ + foreign key table name + primary key table name + foreign key table's constituent field name; table name prefixes may be omitted. **Indexes:** IDX_ + table name + constituent field name. If a composite index has many constituent fields, only the first field is included, with a serial number added; table name prefixes may be omitted.

3.2 Logical Data Partitioning Based on the logical partitioning of Package 07 functional data, tables are categorized as follows: **TI_** (TableInfo_ abbreviation) for parameter-setting related tables, such as manuscript genre, source, and classification. **TB_** (TableBase_ abbreviation) for basic information setting tables, such as departments, personnel, authors, roles, and sharing scopes. **TW_** (TableWork_ abbreviation) for Package 07 business-related tables, such as manuscripts, topics, tasks, and selections. **TR_** (TableRole_ abbreviation) for role permission-related tables.

3.3 Data Association Design Due to numerous association query requirements in Package 07 business, it is necessary to properly handle many-to-many relationships during initial design. Whenever possible, many-to-many relationships should be eliminated by converting them into two one-to-many relation-

ships. For example, a person may have multiple roles, and conversely, a role may be assigned to multiple people. By adding a role-personnel table between the role and personnel tables, the relationship becomes two one-to-many relationships [Figure 1: see original paper]. Furthermore, for key data that may change due to user-specific requirements, a "name-value table" design is added. As the name suggests, when keys are modified by values associated with other data, this design prevents data storage confusion and facilitates multi-table association queries. Third, to achieve data integrity, data redundancy is considered during table design, along with the addition of transactions, cascade deletes, and cascade updates.

Following these rules, indexes are added to key columns in Package 07's manuscript tables, log tables, operation history tables, etc. The optimized retrieval performance is shown in the table below:

Publication	Manuscript Table Records	Log Table Records	Query Time (Before)	Query Time (After)
Peninsula City Daily	300,000+	1,800,000+	5 minutes	1-3 seconds
Dazhong Daily	400,000+	2,000,000+	5 minutes	1-3 seconds

3.4 Future Expansion The database design for Package 07 not only meets current functional requirements but also considers future functional expansion to increase flexibility. **Reserved data tables** are designed for potential future business needs, such as audio-video information extension tables to record bitrate, duration, and other information. **Reserved fields** are also included—most system business data tables reserve 1-2 fields to facilitate functional expansion and subsequent business development. If these reserved fields cannot meet future business needs, additional data tables can be added to achieve higher scalability requirements.

3.5 Optimization Design For large database tables, proper indexing can significantly improve overall database operation efficiency. Indexes constitute a major component of database optimization. When designing data tables, it is crucial to consider that when a single table contains large data volumes—for example, reaching millions of records—ordinary query statements consume considerable time. Creating appropriate indexes on such tables can improve retrieval efficiency by dozens or even hundreds of times. The following principles guide index design: First, primary key columns (sguid) and foreign key columns (fk_{sguid}) must be indexed. Second, frequently queried data columns should be indexed as needed. Third, data columns requiring rapid or frequent range queries, such as "name LIKE 'a%'," should preferably be indexed. Fourth, data columns frequently used in WHERE clauses should be indexed. Fifth,

fields that frequently appear after ORDER BY, GROUP BY, and DISTINCT keywords should be indexed. When creating composite indexes, the field order should match the order of fields following these keywords; otherwise, the index will not be utilized. Sixth, in composite indexes, records are first sorted by the initial field. For records with identical values in the first field, the system then sorts by the second field, and so forth. Only when the first field of a composite index appears in query conditions can the index be used. Therefore, placing frequently used fields at the beginning of composite indexes maximizes the likelihood of index utilization. Seventh, columns rarely involved in queries or with many duplicate values should not be indexed. Eighth, columns defined as text, image, blob, or bit data types should not be indexed. Ninth, columns that undergo frequent updates should avoid indexing. Tenth, limit the number of indexes on a table. For tables with extensive update operations, the number of indexes should generally not exceed three, with a maximum of five. While indexes improve access speed, excessive indexes affect data update operations. Additionally, in Join queries, indexes should be created on the joined fields in both tables.

4. Conclusion

This paper discusses existing problems in some application systems to highlight the importance of database design. It proposes principles and specifications to follow when designing the Package 07 database, thereby creating a high-performance, maintainable, and extensible database that meets Package 07's functional requirements and future expansion needs.

Funding: This research is supported by the National Digital Composite Publishing System Engineering—Media Collaboration System (1141STC40519/07) project.

References: [1] Li Wenjie. Optimization Design Scheme for Large Oracle Databases[J]. Technology Wind, 2011(19). [2] Zhao Jingyu. Theoretical Research and Application of Database Design Standardization[J]. Electronic Technology and Software Engineering, 2013(21). [3] Wang Hui. Design and Implementation of Big Data Application System Architecture[J]. Electronic Technology and Software Engineering, 2015(20).

Author Affiliations: Weifang Beida Jade Bird Huaguang Phototypesetting Co., Ltd.; Dongying Daily News

Note: Figure translations are in progress. See original paper for figures.