

Peer Assessment for DKU Events Database*

Shaoxiang Qu¹ and Qixuan Yuan²

¹ Duke Kunshan University, Suzhou 215316, China
shaoxiang.qu@dukekunshan.edu.cn

² Duke Kunshan University, Suzhou 215316, China
qixuan.yuan@dukekunshan.edu.cn

1 Strengths

1.1 Clear Relational Structure

The relational database schema clear shows the relationship between different tables. With the core of "Events", the database derives many other related tables, such as Users, Attendees and Organizers. As we can see in the schema, there are few intersections of lines in the graph, which means that the data redundancy is reduced and consistency is improved.

1.2 Support for Dynamic User Preferences

The highlight of the database, which is also the group's emphasis, is the application of JSON in the database design. Categories of interests of users, the clubs the user belongs to and the users' preferences to the clubs are store in JSON format. What's more, they use procedures and triggers to automatically disassemble the JSON and insert it into multiple mapping tables. Therefore, it makes the system highly flexible to user preferences.

1.3 High Automation via Procedures and Triggers

The structure of "procedures in Procedure" is also a highlight catching audiences' eyes. In the main procedure, there are five other procedures, which all return their own results such as `_temp_date_result`, `_temp_cat_result`. Such structures, as well as many triggers, achieve high level of automation. For example, when a new user registers, the AFTER INSERT trigger automatically calls the `sp_process_user_preferences`.

1.4 Advanced Filtering System

The `filter_events` process is complex but powerful. It supports filtering by date range, category, organizer, location, event type... Apart from that, dynamic intersect results using multiple temporary tables. There are also many useful views for organizer analytics, like most popular events.

* Emily Biaz, Elliot Aung, Shirley Xiao

2 Weaknesses

2.1 JSON Usage Increases Complexity

Admittedly, as mentioned above, JSON is an interesting part of the database. However, it still shows some disadvantages. First, in the MySQL, the performance of JSON operations may be not as well as expected. Besides, the Loop structures, combined with creating temporary tables, is likely to increase the cost of the query and operations in the database. And if the structure of JSON needs changing, it is difficult to maintain the functions related. Therefore, the usage of JSON can be both a highlight and challenge.

2.2 Possible Low Performance Facing Large Data Volume

Because of the multi-level loops, the procedures related to JSON is likely to have a high time complexity. Therefore, when the data volume is larger, the query and filter time may increase rapidly, which directly leads to low performance.