International Journal of Engineering Research and Development
e- ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 22, Issue 9 (September 2026), PP 78-82

Database Design and Implementation for an Intelligent Tutoring Service and Risk Management Platform

Leqian Ni
E-Commerce Program, Shenzhen Campus, Jinan University, Shenzhen 518053, China
Corresponding Author: nileqian@stu2024.jnu.edu.cn

ABSTRACT
This paper proposes and implements an intelligent tutoring service and risk management platform to solve the problems of online tutoring services, such as irregular verification of tutor credentials, inefficiency of matching supply and demand, mess of contract and evaluation documents, and difficulty in tracking risk-related or non-compliant behavior. The system is based on Python 3.10, Flask and MySQL 8.0 to build a browser-server (B/S) structure. The system aims at three types of users including parents, tutors and administrators The system creates nine basic data tables including subjects, users, requests, matches, contracts, evaluations, risk alerts, and operational logs. By introducing primary and foreign key constraints, transaction management, status synchronization and audit logging, the platform has implemented a complete business process consisting of qualification assessment, request submission, intelligent matching, contract execution, dispute handling and fraud management. Functional testing is done to make sure that each module is able to perform the data operations as specified and errors are handled properly and business information is flowing consistently via the front end, back end and database. The design uses a relational database to verify the feasibility of digitalization of tutoring services and closed-loop risk management.

Keywords: *Intelligent tutoring services; Relational database management; Intelligent matching; Risk mitigation; Audit logging*

Date of Submission: 02-09-2026 Date of acceptance: 12-09-2026

I. INTRODUCTION

Digital tutoring platforms connect the needs of parents with the supply of college student tutors, but their operations include identity verification, matching supply and demand, contract execution, resolution of evaluation disputes, and protection of personal information. Conventional platforms rely on manual verification and real-time communication, making it difficult to uniformly validate instructors' skills, resulting in higher screening costs for parents and not maintaining consistent documentation of class schedules and subsequent reviews. Contract breaches, negative reviews, or perhaps abusive appeals, all of which hinder the evaluation of accountability and risk management. In addition to publishing information and processing orders, the platform links business records at the database level so that they can be traced and verified.

Relational databases ensure consistency of multi-party data through standardized table structures, integrity constraints and transaction procedures. Access control, logging and security evaluation provide the basis for risk detection in database applications. Based on the above standards, this article introduces a database system that integrates business operations and risk management, taking coaching services as an example. The research aims at: 1. Developing a data model that contains three types of users and the full order process. 2. Implementing state synchronization for demand matching, contract signing, payment settlement, review and complaint, and risk alert. 3. Improving the traceability of key operations by separating permissions and logging operations. 4. Completing the integration of front-end and back-end and testing the key business functions. In this article we emphasize the need of reviewing system architecture and the results of implementation at the course design stage, without making any generalizations about large-scale concurrent performance or the effectiveness of recommendation algorithms in real market situations.

II. MATERIAL AND METHODS

The presentation layer consists of the parent console, tutor dashboard, administrator backend and public integrity bulletin board. The business layer is based on Python 3.10 and Flask, and is responsible for realizing the functions of authentication, form verification, matching calculation, contract generation, evaluation of complaints and risk management. The data layer utilizes MySQL 8.0 for data storage and DBeaver for conceptual and physical model design. The business layer is responsible for checking front-end requests and converting them into parametric SQL queries; the results returned by the database will be rendered on the page,

and the system prohibits direct access to the underlying data.

The database adopts the third paradigm design and contains nine main data tables: 'subject_map' stores the basic information of the subject; 'parent' and 'tutor' store the accounts and personal information of parents and tutors respectively; 'demand' stores tutoring subjects, grades , information such as budget, address and demand status; 'match_record' stores the matching score and processing results between the request and the potential tutor; 'contract' stores the class time, pricing, payment and performance details after successful matching; 'feedback' stores the score , evaluation, refund status and appeal status; 'risk_warn' stores the type of violation, rectification requirements and disposal results; 'operation_log' stores the operator, role, operation behavior, IP address and time stamp.Each entity is associated with the foreign key through the primary key, and defines the following data flow: parents submit requirements -> system matching -> both parties sign contracts -> course end evaluation; if there is a violation, the risk management process will be triggered.

The system has built a set of integrity constraint mechanisms, which includes unique constraints, non-empty constraints, foreign key constraints and state constraints. The developer will index the two fields of account identification and mobile phone number respectively. The system will check whether the variables that support the core business operation are empty, and the correlation between application records, matching records, contract records and evaluation records generated by business personnel in the business process will be maintained through foreign keys. Protect. Developers rely on transaction control capabilities to support contract execution, business evaluation and risk management. Developers deploy this capability in multiple interrelated data tables; when the system detects the failure of any key step, it can roll back the completed operation to avoid data inconsistency caused by local data updates. Developers will choose restricted deletion or cascade deletion policies for deletion operations according to the rules of the business logic itself, which can prevent the system from generating isolated data that cannot be associated with other business data. High-risk operations are logged in the 'operation_log' table. The system will also update the mentor's reputation score, warning status and blacklist status according to the tutor's violation record, negative evaluation and case management results.

During the matching process, the system screens potential tutors based on the requirements submitted by parents and comprehensively considers the subject, grade, budget, tutor major, GPA, evaluation, account status and other factors. Then it will calculate the overall game scores and rank them. The matching results are stored in the 'match_record' table. After the order is confirmed, the system will create the corresponding contract in the 'contract' table. This method is used in curriculum design projects, which aims to make the matching process clear and easy to understand, and does not use a recommendation model driven by learning data. This method is used in curriculum design projects, which aims to make the matching process clear and easy to understand, and does not use a recommendation model driven by learning data. Functional tests cover user registration and login, qualification verification, CRUD operations on demand, matching and contract generation, payment processing, attendance records, notification dispute handling, account warning and suspension, and log query. In addition, the test also simulated abnormal situations such as empty values, duplicate accounts, unauthorized access, invalid status changes, and deletion of associated data to verify whether the form verification, database constraints and transaction rollback functions are normal.

III. ICASE STUDY: END-TO-END TUTORING SERVICE AND RISK MANAGEMENT WORKFLOW

The researchers ran a complete tutoring service process on the prototype platform to check whether the function of the designed database is normal. The case study focused on the main stakeholders: parents, tutors and administrators and followed the life cycle of a tutoring request from the establishment of an account, through the provision of the service and ending with a risk assessment. The intent was to determine if the nine-table relational architecture could support consistent business records for the different roles and support intelligent matching, contract management, evaluation processing, risk notifications and audit documentation. The results show that the case study pays more attention to the correctness of the operation and the consistency of the data than high performance and precise alignment.

The process began when a guardian created an account and submitted a tutoring request that included the subject, grade of kid, budget, preferred frequency of teaching, address, phone number and educational requirements. Before storing the request, the business layer will verify the validity of the required fields and the input form. The 'demand' database had a valid request with state pending-matching. The system then joined the request with the 'subject_map' and 'tutor' tables to consider potential tutors based on subject match, academic qualifications, GPA, hourly fee and account standing. The system calculates the matching score for the candidate tutors and sorts them by score. The recommended results are saved in the 'match_record' table, and data can be extracted directly from these records if necessary without specific interface operations.

After the tutor accepts the request, the system will update the matching status corresponding to the request and create a new record in the 'contract' table. This record will save the teaching time, fee, total amount payable, payment method, service details, attendance and tutors assigned by the platform. The foreign key will

bind this contract to the corresponding matching record, which is associated with the parents who initiated the needs, the tutor who undertakes the tutoring, and the original request originally submitted by the user. Follow-up operations such as payment verification and attendance verification will modify existing contracts, and the system will not add new contract records. After the tutoring service, parents can submit scores, comments or refund applications, and the content submitted by parents will be stored in the 'feedback' form. The tutor can file a complaint against this user's evaluation. Relevant service details and dispute records will be retrieved and reviewed by the administrator.

When the administrator finds out that the qualification is wrong, breach of contract or other risk events, the system will generate a record in the 'risk_warn' table. This record marks the tutor involved, and saves the type of violation, internal violation ID, reason for violation, rectification requirements and record creation time. The system can freeze the account if necessary, and the status of the account in the blacklist will also be updated synchronously. The warning log retains the behavioral trajectory left by multiple violations, and the administrator can check these logs. The public page for ordinary users only shows the necessary account status and warning tips, and the complete event record can be retrieved by the administrator in the background. This permission design allows the entire disposal process to be traced back and restricts the disclosure of personal information.

Functional testing covers a complete set of processes, which includes user registration, qualification submission, tutoring demand release, demand matching, contract signing, payment status update and course login, as well as functions such as evaluation, appeal, risk warning, account freezing and administrator audit. The prototype system can perform preset data operations in these business processes. The system relies on unique constraints to prevent duplicate user names, and the business logic layer intercepts over-authorized operations and incomplete submission requests with incomplete information. The correlation between foreign key maintenance requirements, matching records, contracts and evaluations. The transaction management and rollback mechanism prevents incomplete data updates when the multi-step operation fails. Operations such as logging, modifying records, handling complaints, and freezing accounts that need to focus on retaining traces will generate logs in the 'operation_log' data table, and the log records the operator, role, operation behavior, IP address and operation time.

The case analysis shows that this relational data structure connects tutoring demand processing with risk management, forming a complete business closed loop. And it showed that intelligent matching and risk management might be on the same data basis but have different functions. However, the above results are limited to the evaluation of prototype functions in the context of curriculum design. They do not exhibit effectiveness in high-concurrency performance, suggestion accuracy in a real marketplace, or the validity of risk laws when applied to a large amount of behavioral data. Evaluation of these aspects depends on production environment data, mandatory specifications and broader operational review.

IV. RESULTS

The system has completed main business components for parents, tutors and administrators. Parents can create accounts, submit and track tutoring requests, search for suitable tutors, approve matches, review contracts, make final payments and leave reviews or request refunds. Tutors can update their resumes and qualification information, view and accept matching tutoring tasks, check contracts, record teaching time, and complain about negative reviews. Admin can view the tutor's credentials, accept and reject requests and agreements, appeals and notifications. Suspend the account of the users who have less credit. The public bulletin board collects the information from the tutor table and the risk_warn table to display excellent tutors, violation records and blacklist criteria.

Continuous processing of the above stated business operations is supported by nine data tables. After parents submit the requirements, the system will deposit them in the 'demand' table, and then filter the suitable tutor according to the information in the 'subject_map' and 'tutor' tables, and save the matching results in the 'match_record' table. After the review is approved, the system will create a "contract" form. After the course is completed, the 'feedback' table is connected to the contract, parent, and tutor. The risk management agreement will lead to inconsistencies in the evaluation or contract performance. The "Risk Warning" table stores the warning object, violation category, description and remediation status, and the administrator's solution will update the mentor's credibility and blacklist entries synchronously. The "operation log" table records the operator, role, timestamp and client IP address of important events such as login, personal information modification, order processing and account suspension, so as to provide searchable audit tracking of company status changes.

During the functional test, the standard input basically performs the specified CRUD (create, read, update, delete) activities, and the page presentation reflects the database entries. Unique constraints do not allow duplicate accounts. Submissions are rejected if not filling mandatory fields. Tutors who fail the review process or the account is suspended will not be able to enter the standard order acceptance process. Non-existent foreign

key entities cannot create valid matches or contracts. Restrict cross-role access by checking permissions. The collaborative actions such as the contracts, assessments and risk evaluations can be reversed if an anomaly exists and the incomplete data was not included into the case.

The inspection also gave more evidence of the effectiveness of the risk management procedure. The administrator can set an alert and suspend the account if they find a discrepancy in the qualifications during the evaluation. When a rating dispute is raised, a pending arbitration entry will be created, and the result will be displayed in the rating status and the status of the tutor. Accounts with repeated violations can be seen from the public notice board. Operational logs provide the historical recording of important actions by account, role and time. The results show that the system has formed a simple closed-loop process, covering four links: service matching, performance documents, dispute handling and risk monitoring. However, this evaluation mainly revolves around the curriculum design. The purpose is to verify whether the data structure and functional scenarios are running. It does not introduce large-scale user samples, nor does it provide quantitative indicators such as concurrent load, response time distribution, and matching accuracy.

V. DISCUSSION AND CONCLUSION

The core value of this system is to integrate the coaching service module and risk management capabilities by relying on standardized relational data structure. Daily business relies on records such as demand, matching, contract and evaluation, and risk warning and operation logs will retain abnormal events and the disposal process corresponding to the event. These two types of data are associated with the tutor number, parent number and contract number. Administrators can use these numbers to trace the business results to the corresponding business subject and operation documents. Compared with systems that only have information distribution functions, this architecture is more effective in maintaining state consistency and providing databases for conflict resolution and account supervision.

We encountered three major problems in the design stage. First of all, the failure of any process in the large business chain may lead to inconsistency or contradiction of information, so the system uses foreign keys, transactions and status verification to control the update parameters. Secondly, multi-condition matching requires a trade-off between interpretability and query efficiency. Although ordered results can be obtained through thematic mapping and systematic condition screening, the validity of its recommended results has not been verified based on real supply and demand data. Most of the current risk management rules are based on manually set restrictions and administrative supervision, which cannot replace real identity verification, legal review and expert credit assessment.

From the perspective of database architecture, there is still room for improvement in the system. At present, the credibility rating of the tutor is directly adjusted according to specific business events; however, although this method is simple, it may lead to inconsistency of historical data when regulations change. In the future, risk events, score changes and adjustment reasons should be recorded separately. The credibility rating should be calculated regularly according to the details of the incident, and the version of the rules should be recorded for audit tracking. For sensitive information such as identity document path, phone number and home address, encrypted storage, data desensitization and access restrictions should be provided; do not directly record complete privacy data. The technical team can establish a set of composite indexes, which is based on business themes, account status and budget intervals to improve the efficiency of matching queries. The team can verify the effect of optimization by performing executive plan analysis and stress tests. Before the project is deployed to the production environment, it needs to clearly formulate the data retention cycle, the user authorization withdrawal mechanism, the appeal review process and the administrator operation approval process to prevent confusion between technical traceability and business compliance verification.

This relational model composed of nine data tables provides functions such as user management, demand submission, intelligent matching, contract execution, dispute determination, risk notification and operation management. The results of the functional test prove that the system can achieve a closed loop of basic business including abnormal detection. The follow-up work can be divided into several directions: the project team uses real authorization data to evaluate the order acceptance rate, response time and concurrent performance; new indicators such as geographical distance, fulfillment rate and historical preference of system access; the team strengthens sensitive data encryption, cloud storage, grading permissions and customizable risk management protocols. Further verify the engineering reasonableness of the system.

REFERENCES

[1]. Li, J., Wang, K., Chen, Y., Zhou, Y., Wu, L., and Wang, J. 2025. Detecting DBMS bugs with context-sensitive instantiation and multi-plan execution. Computers & Security, 157, 104564. https://doi.org/10.1016/j.cose.2025.104564.

[2]. Zhou, X., Chai, C., Li, G., and Sun, J. 2022. Database meets artificial intelligence: A survey. IEEE Transactions on Knowledge and Data Engineering, 34(3), 1096 – 1116. https://doi.org/10.1109/TKDE.2020.2994641.

[3]. Muñoz-Arcentales, A., López-Pernas, S., Pozo, A., Alonso, Á., Salvachúa, J., and Huecas, G. 2019. An architecture for providing data usage and access control in data sharing ecosystems. Procedia Computer Science, 160, 590 – 597.

https://doi.org/10.1016/j.procs.2019.11.042.
[4]. Qin, X., Huang, Y., Yang, Z., and Li, X. 2021. A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing. Journal of Systems Architecture, 112, 101854. https://doi.org/10.1016/j.sysarc.2020.101854.
[5]. Zhao, X., Wang, S., Zhang, Y., and Wang, Y. 2022. Attribute-based access control scheme for data sharing on Hyperledger Fabric. Journal of Information Security and Applications, 67, 103182. https://doi.org/10.1016/j.jisa.2022.103182.
[6]. Qiao, S.-J., Fan, H.-L., Han, N., Du, L., Peng, Y.-H., Tang, R.-M., and Qin, X. 2025. Learning database optimization techniques: The state-of-the-art and prospects. Frontiers of Computer Science, 19(12), 1912612. https://doi.org/10.1007/s11704-025-41116-7.