Exploration on Teaching Reform for the" Database Principles and Applications " Course Based on CDIO-OBE Concept

Qing Qu [1, a,*], Yuxin Li [1, b]

[1]Xi'an Technology and Business College, Xi'an, shaanxi, China

[a] 1451129680@qq.com, [b] 2858436721@qq.com

Abstract. Aiming at the problems existing in the traditional teaching of "Principles and Applications of Database Systems" for software engineering majors in private application-oriented undergraduate universities—such as overemphasis on theory while neglecting practice, significant differences in students' foundational knowledge, and a single evaluation method—this study integrated the CDIO (Conceive-Design-Implement-Operate) engineering education model with the OBE (Outcome-Based Education) concept, and implemented a systematic teaching reform using the "Student Performance Management System" as the core teaching case. Following the CDIO process of "Conceive-Design-Implement-Operate" and relying on SQL Server as the database tool, student-centered project-based teaching was carried out. Adhering to the logic of "reverse design and forward implementation", a three-dimensional objective system covering knowledge, competence, and quality was reconstructed. Progressive project-based teaching was adopted, along with a differentiated strategy of "stratified grouping + flexible tasks", and a multi-dimensional assessment system consisting of "process evaluation (60%) + summative evaluation (40%)" was established. Teaching practice shows that this model effectively stimulates students' learning initiative, and significantly enhances their engineering practice and teamwork abilities. The CDIO-OBE integrated model can effectively improve the teaching quality of application-oriented courses in private universities, providing a referable path for the reform of similar courses.

Keywords: CDIO; OBE; Database Principles; Applications; Teaching Reform; Diversified Assessment.

1. Introduction

Outcome-Based Education (OBE) is an educational model designed to cultivate the knowledge, skills, and qualities students need to achieve success. CDIO (Conceive-Design-Implement-Operate) represents a new engineering education philosophy and implementation system focused on student engineering practice, serving as a highly effective engineering education model for enhancing engineers' capabilities [1].

With the rapid advancement of information technology, enterprises' demand for the capabilities of fresh graduates in software engineering has shifted from single-skill proficiency to comprehensive competence. Private universities serve as vital training grounds for applied talent. Reforming curricula to enhance graduates' job readiness is crucial, enabling them to handle database development and maintenance tasks for small and medium-sized enterprises. However, traditional teaching models face certain challenges in private university settings. Disconnect between theoretical instruction and practical application: Classroom focus often centers on abstract concepts like relational algebra and normalization theory, while lab sessions typically involve repetitive, verification-based SQL operations. The absence of comprehensive, project-based learning throughout the curriculum hinders students' ability to translate theoretical knowledge into practical engineering skills. Traditional teaching methods often rely heavily on instructor-led lectures with limited interaction, resulting in low student engagement and diminished interest—particularly in the more theoretical and potentially tedious aspects of database theory. This approach can foster a sense of intimidation among students. Furthermore, conventional assessment methods predominantly emphasize knowledge recall, which may not align with the course's learning objectives, assessment priorities, and engineering characteristics. Neglecting students' varying foundational skills: Students at private universities possess uneven computer literacy, making it difficult for standardized teaching

https://doi.org/10.58244/joerai.263696

pacing and content to accommodate learners at different proficiency levels. Relying on a single evaluation method: Over dependence on final written exams overlooks the assessment of teamwork, innovative thinking, and engineering practice skills during the learning process [2-3].

To address these issues, scholars worldwide are actively exploring the application of CDIO and OBE principles in engineering education. Multiple institutions have achieved a shift from "teacher-centered" to "student-centered" education through OBE-driven curriculum restructuring [4-6]. However, current research predominantly focuses on applying single methodologies, with limited systematic exploration of the deep integration of CDIO and OBE. Particularly in database principles and applications courses at private universities, tiered instructional design addressing student skill disparities and project-based assessment systems remain underdeveloped.

Based on the CDIO-OBE integration concept, the innovations in the Database Principles and Applications course are as follows:

(1) Tiered Instructional Design: To address variations in students' foundational abilities, a tiered in-class assessment mechanism (covering basic and advanced levels) has been designed to facilitate skill enhancement.

(2) Project-Based Practice Integration: Using a "Student Grade Management System" as the core case study, knowledge points such as E-R diagram design and normalization are integrated into the four CDIO phases, enhancing knowledge coherence and practical applicability.

(3) Diversified Evaluation System: A quantitative framework combining "Formative Assessment (60%) + Summative Assessment (40%)" is established to comprehensively evaluate student performance across all project stages.

2. Current Status and Analysis of the "Database Principles and Applications" Course

2.1 Course Positioning and Characteristics.

"Database Principles and Applications" serves as a core course in software engineering programs at private institutions, occupying a pivotal bridging role. It not only connects foundational courses like "Programming Fundamentals" with subsequent courses such as "Web Development" and "Big Data Applications," but also functions as a key vehicle for cultivating students' abilities in data modeling, SQL programming, and system optimization [7].The core characteristics of this course are reflected in: Theoretical Level: Relational algebra and normalization theory form the mathematical foundation of database design, requiring students to possess strong logical abstraction abilities. Practical Level: Students must master tools like SQL Server to complete end-to-end training—from requirements analysis and E-R modeling to SQL implementation and system optimization—cultivating an engineering mindset [8]. However, this dual emphasis on theory and practice also increases teaching complexity: abstract concepts can intimidate students, while practical sessions lacking systematic project guidance risk becoming fragmented operational drills.

2.2 Analysis of the Teaching Audience.

For the 2023 cohort of the Software Engineering program at this institution, comprising 4 classes totaling 139 students. Along with comparable domestic universities, revealed that students at private institutions typically exhibit "strong practical interest but weak theoretical foundations." Their strengths include high enthusiasm for hands-on activities and the ability to rapidly acquire foundational skills through case-based learning. Weaknesses include poor mathematical foundations, with approximately 40% of students struggling in areas like function dependency derivation and relational algebra operations; limited teamwork experience; and frequent imbalances in task distribution during project practice. These characteristics necessitate a teaching model that integrates differentiated instruction with systematic development of team collaboration skills.

2.3 Necessity of Reform.

To address these issues, the integrated concept of "Outcomes-Based Education (OBE) + Capability Development through Industry-Oriented Education (CDIO)" is introduced. Its necessity manifests in three key aspects:(1) Aligning with Industry's Full-Process Competency Demands: Enterprises require database professionals to possess end-to-end capabilities spanning "requirements analysis-design-implementation-operation and maintenance." Traditional teaching emphasizes the

https://doi.org/10.58244/joerai.263696

"implementation" phase, whereas CDIO-OBE employs real projects to integrate all four stages, enabling students to experience the complete workflow. (2) Addressing Varied Student Foundations: Students at private institutions exhibit significant disparities in foundational knowledge. Tiered task design enables participation for learners at all levels, achieving "teaching tailored to individual aptitude." Simultaneously, during group collaboration, instructors can provide personalized guidance based on each student's foundation and abilities, helping them successfully complete tasks while enhancing their sense of learning achievement and confidence. (3) Implementing holistic education goals for new engineering disciplines: Centered on students, CDIO-OBE emphasizes balanced development of competencies and professional ethics. Through group collaboration and project presentations, it cultivates not only specialized knowledge and skills but also enhances communication abilities and professional conduct. These elements align with the new engineering discipline's trinity of goals: value shaping, competency development, and knowledge transmission.

3 Design of Database Course Teaching Reform Based on CDIO-OBE

3.1 Integration of CDIO-OBE Principles and Overall Framework.

Although CDIO and OBE principles differ in emphasis, they exhibit strong complementary integration, forming an organic whole that jointly advances the in-depth development of educational reform. The OBE philosophy provides CDIO with a clear goal orientation. During CDIO implementation, OBE helps educators define the ultimate competency objectives students should achieve—that is, "what competencies to cultivate." By thoroughly analyzing industry demands and occupational standards, specific competencies students must possess are identified, giving CDIO projects a clear direction and purpose.

CDIO, in turn, provides a practical implementation pathway for OBE. Through the full project lifecycle of "Conceive-Design-Implement-Operate," CDIO translates the competency goals from the OBE philosophy into concrete teaching practices. At each project stage, students progressively master and enhance required competencies through hands-on practice and teamwork. This facilitates the transition from theoretical knowledge to practical skills while simultaneously developing their engineering practice and innovation capabilities. Such experiences validate the attainment of competency goals established under the OBE framework [9-10].

The integrated teaching reform model combining Outcome-Based Education (OBE) and Conceptualization-Design-Implementation-Operation (CDIO) principles is illustrated in Figure 1.

3.2Backward Design: Three-Dimensional Instructional Objectives Based on OBE.

Based on enterprise research, the three-dimensional course objectives are designed through backward design:

Knowledge Objectives:

(1) Master fundamental database concepts and principles.

(2) Proficiently operate SQL Server tools and T-SQL programming. Independently complete database creation, table design, data queries, and related operations.

(3) Master the complete database design process from requirements analysis and E-R diagram design to physical implementation.

Competency Objectives: (1) Foundational Competencies: Independently write SQL statements (DQL queries, DML operations, DDL definitions). Achieve ≥90% pass rate on foundational SQL assessments. (2) Advanced Competencies: Master SQL writing techniques. Design E-R diagrams and perform normalization optimization based on system requirements. Translate real-world entities and relationships into conceptual database models. (3) Comprehensive Competency: Collaborate in a team to complete the database design and implementation for a "Student Grade Management System."

Attitude Objectives: (1) Demonstrate teamwork: Learn to collaborate with others in group projects, respect peers' opinions and suggestions, and jointly accomplish project tasks to enhance teamwork skills. (2) Cultivate professional ethics: Adhere to coding standards, maintain complete documentation, develop rigorous engineering thinking, approach problems from an engineering perspective, and prioritize system design, implementation, and maintenance. (3) Cultivating a habit of self-directed learning in new technologies, learning to summarize lessons, continuously improving

https://doi.org/10.58244/joerai.263696

design and implementation solutions, and enhancing professional competence and overall quality.The interrelationship between the key components is illustrated in Figure 1.

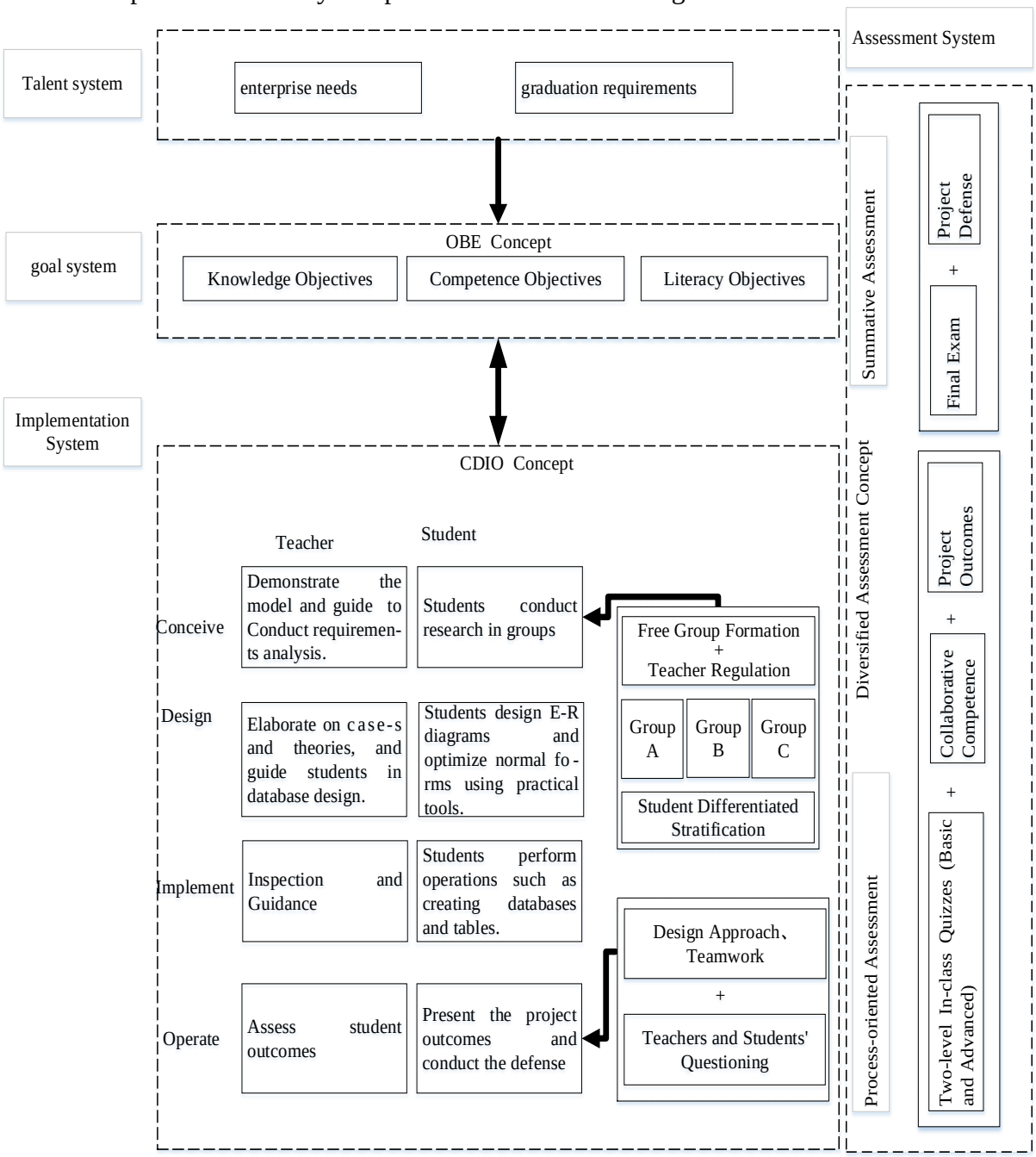

Figure 1: CDIO-OBE Integrated Teaching Reform Framework Diagram

3.3 CDIO-based Progressive Reconstruction of Teaching Content.

Centered on the “Student Grade Management System” project, teaching content is decomposed according to the four CDIO stages into progressive tiers: “Foundation-Intermediate-Comprehensive-Advanced.” This approach accommodates varying student foundations while fully leveraging SQL Server tools to achieve a “theory-practice” closed loop. Tool application details for each stage are as follows:

3.4 Implementation of Differentiated Teaching and Process - oriented Assessment Strategy.

(1) Design of Student Stratification and Flexible Tasks. Based on students' prior coursework performance and varying foundational knowledge, a “stratified grouping + flexible tasks” strategy is adopted. Students are divided into three tiers for differentiated instruction:The specific strategies for curriculum design and learner grouping are detailed as illustrated in Table 1 (CDIO-based Progressive Teaching Content Design) and Table 2 (Student Stratification Based on the Implementation of

Differentiated Instruction).

Table 1. CDIO-based Progressive Teaching Content Design

Teaching Stage	CDIO Phase	Core Applications of SQL Server Tools	Students' Main Operational Tasks
Basic Stage	Conceive	Create databases via SSMS graphical interface and T-SQL statements; Define table field types using Table Designer; Add primary keys and check constraints with Constraint Manager.	Independently complete database creation and table structure design, and submit screenshots; Write CREATE TABLE statements with T-SQL (compare differences between graphical and code operations); Insert test data.
Advanced Stage	Design	Draw system E-R diagrams; Configure foreign key relationships in Table Designer; Create "score statistics views" via View Designer; Perform multi-table join queries.	Independently draw E-R diagrams and correct foreign key association errors; Write and query views using T-SQL; Complete normalization determination to avoid data redundancy.
Comprehensive Stage	Implement	Develop core system functions; Write triggers to implement business logic.	Develop core functions; Write triggers and test trigger functions.
Enhancement Stage	Operate	Perform user permission management; Prepare for project defense.	Project defense: demonstrate system functions, explain problem-solving processes, and propose improvement directions.

Table 2: Student Stratification Based on the Implementation of Differentiated Instruction

Stratification	Differences in Core Tasks
Group A (Weak Foundation)	Focus on consolidating basic skills. Tasks: Complete table creation and single-table queries with reference to templates; imitate and write multi-table join queries.
Group B (Medium Foundation)	Focus on skill application and transfer. Tasks: Independently complete table creation and multi-table queries; draw preliminary E-R diagrams; independently develop simple stored procedures.
Group C (Strong Foundation)	Focus on system design and optimization. Tasks: Independently complete database design, including E-R diagram optimization and normalization determination; develop complex triggers.

(2) Design and Implementation Effect of In-class Quizzes.

To assess learning outcomes in real time, two in-class quizzes closely integrated with projects are designed:

https://doi.org/10.58244/joerai.263696

Quiz 1 (Basic SQL) Core Topics: Create a “Student Table” and insert data; Query “Names and Student IDs of Male Students in Class of 2024”; Update “Zhang San's Grade to 2023”; Delete “Course with ID C001.”
Quiz 2 (Advanced SQL): Core topics: Multi-table join query for “student names, course names, and scores”; subquery to “identify student IDs and scores exceeding the average”; create a view for “Grade 2024 student performance statistics (including average scores)”.
(3) Group Collaboration and Role Rotation.
Employ a “free grouping + teacher fine-tuning” approach to ensure every student experiences the full CDIO process and develops comprehensive skills.
Self-Formation Phase: Groups of 3-4 students, ensuring each includes A, B, and C-level students to achieve “stronger students guiding weaker ones.”
Faculty Adjustment Phase: Groups are restructured to ultimately meet the composition of “1 Group C student + 1-2 Group B students + 1 Group A student.”
Role Rotation Mechanism: Clearly defined roles are rotated to ensure each student participates in the entire process, covering the full cycle of “Requirement Analysis - Design - Development - Testing.”

3.5 Establishing a Diversified Assessment System.

Moving beyond the “single final exam” model, we implement a “formative assessment (60%) + summative assessment (40%)” system that balances “knowledge acquisition, skill development, and character cultivation”:

Table 3: Diversified Assessment System

Evaluation Dimension	Evaluation Content and Method	Evaluation Subject	Weight
Process-oriented	Two in-class quizzes; phased project outcomes (requirements documents, E-R diagrams, SQL code); group collaboration performance (contribution degree, communication ability).	Teachers, group leaders	60%
Summative	Final written exam (focusing on comprehensive application); final project defense (system functions, design approach).	Teachers	40%

4 Teaching Reform Practice: CDIO-OBE-Based Instructional Implementation Process

4.1 CDIO Four-Phase Implementation Driven by Projects.

Using the “Student Grade Management System” as the vehicle, fully implement the CDIO engineering education model.
(1) Conception Phase (Conceive): Demand-Driven, Goal Clarification.
Task Objective: At the course outset, instructors demonstrate the “Student Grade Management System” prototype through hands-on operation, showcasing system functions such as student information entry, query, and modification; grade input, statistics, and analysis; and report generation. This provides students with an intuitive understanding of the system, its practical application scenarios, and value, stimulating learning interest and motivation. Instructors guide students to analyze the system's functional requirements, prompting them to consider implementation methods and data support for each module. This introduces the course content and lays the foundation for subsequent learning activities.
Implementation Steps: Instructors provide a “Requirement Research Template.” Students conduct interviews in groups, engaging with faculty and peers to understand functional requirements and expectations for the “Student Grade Management System.” Through group discussions, they consolidate these requirements into a “Requirement Specification Document.” During drafting, students must clearly define the system's functional and performance requirements, providing explicit guidance for subsequent design and implementation phases.

https://doi.org/10.58244/joerai.263696

(2) Design Phase (Design): Entity-Relationship Diagrams and Normal Form Optimization, Aligned with Job Design Competencies.

Task Objective: Transform requirements into conceptual and logical models, identify and optimize normal forms to ensure data integrity.

Implementation Steps: Teams use tools like Visio to draw E-R diagrams, clarifying relationships between entities; instructors use specific cases to help students master methods for determining 1NF-3NF. After explaining normalization theory, instructors guide students through normalization analysis. Through group discussions, students complete normalized relational schema designs and submit a Database Design Report.。

(3) Implementation Phase（Implement）: Layered Practice to Strengthen Skills.

Task Objective: Implement database designs in SQL Server, develop core functionalities, and complete testing.

Implementation Steps: Basic operations: Write DDL statements to create databases and tables, set primary keys, foreign keys, and constraints; Advanced operations: Write views, stored procedures, and triggers; Functional testing: Develop test cases to validate functionality and data integrity.

Instructors provide on-site guidance and deliver focused lectures on common challenges like “errors in multi-table join queries.”。

(4)Operation Phase（Operate）: Present Project Outcomes.

Task Objectives: Optimize database performance, showcase project deliverables, summarize lessons learned, and conduct a final presentation.

Implementation Steps: Comprehensively test the functionality of the “Student Grade Management System.” Develop test cases to verify whether the system correctly handles concurrent operations, ensuring data consistency and integrity.

Each group presents for 15 minutes. During the defense, emphasize explaining design concepts—such as E-R diagram design, normalization principles, trigger application, and team collaboration processes—while answering questions from instructors and peers. Defense scores are incorporated into the final evaluation.

4.2 Dynamic Feedback Mechanism.

At the conclusion of each phase, instructors publish a Project Progress Report providing detailed analysis of each group's performance in areas such as E-R diagram design and SQL efficiency. The report highlights issues in E-R diagram design—including entity relationship errors and attribute redundancy—and offers improvement suggestions to guide students in optimizing their E-R diagrams and enhancing database design quality. Regarding SQL efficiency, instructors analyze whether teams' SQL statements contain performance bottlenecks—such as full table scans or improper index usage—and offer optimization suggestions like creating appropriate indexes or refining query structures. This dynamic feedback mechanism establishes an “evaluation-feedback-optimization” loop, enabling students to promptly identify issues, adjust learning strategies, and continuously enhance their learning outcomes and professional competencies.

5 Teaching Reflection and Improvement

5.1 Challenges Encountered During Reform.

Teacher Resources and Class Time Constraints: Tiered task design and formative assessment require additional time investment. In project-based learning, students need ample group discussion to jointly analyze problems and explore solutions—a process demanding significant time. Teachers must provide personalized guidance tailored to each group's specific situation, address student inquiries, and assist with problem-solving, further requiring greater time and effort. However, limited class hours struggle to accommodate project-based learning's time demands, preventing certain teaching segments from being thoroughly developed and consequently impacting instructional outcomes.

Low Collaboration Motivation Among Some Students: A minority of Group A students relied excessively on peers while contributing minimally.

Addressing Foundational Gaps: Despite implementing tiered instruction and personalized guidance,

https://doi.org/10.58244/joerai.263696

a small number of students still faced significant challenges in abstract theoretical learning, such as paradigm identification. These students may struggle to grasp the principles and methods of paradigm determination due to weak mathematical foundations or insufficient logical thinking skills. For this group, additional learning resources such as online micro-lectures should be provided, enabling self-paced learning according to individual schedules. Concurrently, one-on-one tutoring should be offered to address learning challenges, ensuring all students master fundamental knowledge and skills.

5.2 Continuous Improvement Measures.

To address the aforementioned issues, the following improvement measures are proposed:

Faculty Support: Establish a "Course Teaching Team" with divided responsibilities for "task design-assessment-question answering" to reduce individual workload; introduce an "automated grading system" to minimize time spent on mechanical grading.

Collaborative Management: Implement a "Contribution Metrics Table" (e.g., SQL code submissions, discussion participation) and mandate weekly "Team Contribution Reports" from each group. Instructors will conduct individual discussions with low-contributing students.

Develop an online micro-lecture resource repository to support personalized learning.

Resource Optimization: Invite industry mentors to participate in project defense evaluations and introduce a case repository featuring scenarios closer to real-world business contexts.

6 Conclusion

This study addresses teaching challenges in the Database Principles and Applications course at private universities by integrating CDIO-OBE principles. It constructs a teaching reform plan based on CDIO-OBE integration, effectively resolving issues such as disconnect between theory and practice, significant variations in student foundations, and monolithic evaluation methods through a reverse-designed goal system, tiered teaching content, differentiated implementation strategies, and a diversified assessment system. This demonstrates the applicability and effectiveness of CDIO-OBE in applied courses at private institutions while fostering teamwork and innovative thinking.

Future efforts will focus on deepening industry-education integration, exploring smart teaching assistance, continuously refining the reform plan, and conducting long-term tracking of graduate development to further validate and enhance this reform model.

References

[1] Zhou Rui, Bao Honghui. Research on Teaching Model for Research-Oriented Graduation Projects in Food Science and Engineering Based on CDIO-OBE [J]. Agricultural Products Processing, 2024, (22): 140-144. 1671-9646(X).2024.22.029. (In Chinese)

[2] Zu Yikang, Xu Miaojing, Xu Chunhui. Research on Teaching Model for Microcontroller Course Design Based on CDIO Concept [J]. Computer Education, 2019(9): 34-37. (In Chinese)

[3] Bai Erjing, Pang Shulan. Research on "OBE-CDIO" Teaching Reform for Applied Undergraduate Computer Majors in the Context of Accreditation Evaluation [J]. Fujian Light Industry and Textile, 2025(1):68-71. (In Chinese)

[4] Wang Junmei, Wu Jihong, Zheng Dongxia, et al. Exploration of Blended Teaching for Software Testing Courses Based on OBE and CDIO [J]. Software Engineering, 2019,22 (10):54-56. (In Chinese)

[5] Lu Dalin, Wu Bin. Tutorial on SQL Server 2008 Database Application and Development [M]. Beijing: Higher Education Press, 2020. (In Chinese)

[6] Zhang Ying, Feng Sang, Liu Yanwei, et al. Exploration of Theoretical Mechanics Teaching Reform Based on OBE-CDIO Integration for Vehicle Engineering [J]. Times Auto, 2025, (07): 74-76. (In Chinese)

[7] Luo Dongmei, He Shanshan, Sun Wenling, et al. Exploration and Practice of Embedded Systems Curriculum Reform Under CDIO-OBE Philosophy [J]. Computer Knowledge and Technology, 2025, 21(08): 161-164. 2025.0406. (In Chinese)

[8] Zhai Mingliang, Xie Jiucheng, Yu Liang, et al. Exploration of Reforming Python Programming Courses Using a Diverse Blended Approach Driven by OBE and CDIO Educational Philosophies [J]. Chinese Character Culture, 2024, (19): 181-183. 2024.19.064. (In Chinese)

[9] Wang Yan, Meng Yakun, Zhang Bin, et al. Research on Project-Driven Teaching Reform of Access Database Courses Based on CDIO [J]. Software, 2020, 41(05): 75-77+142. (In Chinese)

[10] Hu Taoying, Yu Juan. Analysis of Java Programming Course Reform Under CDIO-OBE Engineering Education Philosophy [J]. Computer Knowledge and Technology, 2024, 20(11): 137-139. 2024.0557. (In Chinese)