

Baxış Tahir oğlu CƏFƏROV

Qərbi Kaspi Universitetinin magistrantı, Bakı, Azərbaycan

E-mail: jafarovbakhish@gmail.com

TƏLİMİN İDARƏETMƏ SİSTEMLƏRİNDƏ HƏR CƏHDDƏ TEST ANALİTİKASI: VERİLƏNLƏR BAZASI SXEMİNİN DİZAYNLANMASI VƏ SORĞU OPTİMALİZASIYASI

Xülasə

Bu məqalə, Təlim İdarəetmə Sisteminin (LMS) müəllimlərə mənalı hər cəhd üçün viktorina analitikasını təqdim edib-etmədiyini müəyyən edən verilənlər bazası dizayn qərarlarını araşdırır. Geniş istifadə olunan platformaların - Moodle, Canvas və Google Classroom-un icmalı göstərir ki, hər cəhd üçün məlumat saxlama ya yoxdur, ya konfigurasiyadan asılıdır, ya da kommersiya lisenziyalaşdırma səviyyələrinin arxasında kilidlənir. Məqalədə iddia edilir ki, hər bir viktorina təqdimatını tələbənin əvvəlki balının üzərinə yazmaq əvəzinə, müstəqil verilənlər bazası sətri kimi saxlamaq analitika qabiliyyəti üçün ən vacib sxem qərarıdır. Doqquz cədvəlli əlaqəli sxem təqdim olunur, xüsusilə cəhdlər və cavab cədvəllərinə, analitik sorğuları miqyasda sürətli edən kompozit indekslərə və tətbiq səviyyəsində yenidənqurma olmadan tələbə başına, hər cəhd üçün bölünmələr yaradan SQL birləşmə modelinə diqqət yetirilir. Yanaşma magistr dissertasiyası layihəsi kimi hazırlanmış universitet tədris-iş yükü idarəetmə platforması kontekstində nümayiş etdirilir.

Açar sözlər: öyrənmə analitikası, verilənlər bazası sxeminin dizaynı, relyasiya verilənlər bazası, SQL indeksləşdirmə, LMS.

UOT: 004.65; 37.018.43; 004.42

DOI: <https://doi.org/10.54414/OUTT9390>

Giriş

Təlim İdarəetmə Sistemləri (LMS) — tədris materiallarını, qiymətləndirmələri və tələbə məlumatlarını vahid rəqəmsal mühitdə birləşdirən platformalar kimi — müasir ali təhsilin əsas infrastrukturuna çevrilib [2]. Təlim analitikasının akademik tədqiqatı - öyrənmə mühitini optimallaşdırmaq məqsədilə tələbələr haqqında məlumatların ölçülməsi, toplanması və bildirilməsi - Siemens və Long sahənin əsas təklifini irəli sürdükdən sonra əhəmiyyətli dərəcədə böyüyüb: tələbələrin qiymətləndirmə materialları ilə necə məşğul olduqları haqqında məlumatlar tədris qərarlarını nəticələri yaxşılaşdırma şəkildə dəyişdirə bilər [3, s.30]. Bu iddianın empirik əsası Arnold və Pistillinin Purdue Universitetindəki işi ilə gücləndirilib və bu iş erkən müdaxilə məlumatlarına əsaslanan və yekun kurs qiymətləri arasında statistik cəhətdən əhəmiyyətli bir əlaqəni nümayiş etdirib [4, s.267].

Bu tədqiqat konsensusuna baxmayaraq, sərbəst şəkildə mövcud olan LMS platformalarının əksəriyyəti ədəbiyyatda faydalı hesab edilənlərdən çox aşağı olan analitik imkanlar təklif edir. Google Forms üzərində qurulmuş Google Classroom-un viktorina funksiyası, standart olaraq hər tələbə üçün yalnız ən son təqdimatı qeyd edir. Moodle cəhdlərin izlənməsini dəstəkləyir, lakin diqqətli konfigurasiya tələb edir və bir çox yerləşdirmələrdə əvəz edilmiş cəhdləri ləğv edən standart parametrlərdə qalır. Nəticə budur ki, tələbənin yalnız hansı balı əldə etdiyini deyil, həm də bu balı təkrar imtahanlarda necə dəyişdiyini və tələbənin hansı konkret suallara ardıcıl olaraq səhv cavab verdiyini bilmək istəyən müəllimlər tez-tez bu məlumatı istifadə etdikləri platformadan əldə edə bilmirlər [1, s.18].

Bu məqalə analitika qabiliyyətini müəyyən edən sxem səviyyəli qərarlara diqqət yetirir. Burada iddia olunur ki, sual əsasən

tətbiq məntiqi ilə bağlı deyil - bu, verilənlər bazasının qorumaq üçün nəzərdə tutulduğu suallardan biridir. Tətbiq yalnız sxemin nəyi saxladığını bildirməyə bilər və əvvəlki cəhd məlumatlarını sətir səviyyəsində yenidən yazan sxem müvafiq analitikanı birdəfəlik bərpa olunmaz hala gətirir.

Əsas sxem qərarı: müstəqil sətir və üzərinə yazın. Tələbə ikinci dəfə test təqdim etdikdə, verilənlər bazası dizayn seçimi ilə üzləşir: mövcud sətir yeni balı əks etdirmək üçün yeniləyin və ya bu xüsusi təqdimatı təmsil edən yeni bir sətir əlavə edin. İki yanaşma tələbənin ilk cəhdi anında eyni yaddaş yaradır; onlar ikinci cəhddən etibarən daimi olaraq fərqlənilir.

Üzündən yazma modeli daha sadə və yaddaş baxımından daha səmərəlidir. Hər test üçün hər tələbə üçün tək bir sətir cari balı saxlayır və `WHERE student_id = ? VƏ quiz_id = ?` ilə sadə `SELECT` cavabı qaytarır. Lakin, bu, trayektoriyayı məhv edir. Üç cəhddən sonra verilənlər bazası tələbənin harada olduğunu, lakin necə ora çatdığını deyə bilməz. "Bu tələbənin balı təkrar imtahanlar arasında yaxşılaşdı mı?" sualı cavabsız qalır - tətbiqin onu hesablaya bilməməsi üçün deyil, əsas məlumatların artıq mövcud olmaması üçün [5, s.88].

Müstəqil sətir modeli hər təqdimat üçün cəhdlər cədvəlinə bir sətir əlavə edir. Saxlama dəyəri orta təkrar imtahan sayına mütənasib sabit əmsalla daha yüksəkdir, lakin analitik qabiliyyət keyfiyyətcə fərqlidir. Romero və Ventura trayektoriya məlumatlarının - cəhdlər üzrə balların ardıcılığının - təhsil məlumatlarının çıxarılması modellərində ən proqnozlaşdırıcı xüsusiyyətlərdən biri olduğunu və istənilən tək ballıq baldan daha informativ olduğunu nümayiş etdirdilər [6, s.608]. Arnold və Pistilli oxşar şəkildə erkən cəhd nümunələrinin son nəticələrin cəhd ballarından daha güclü proqnozlaşdırıcı olduğunu aşkar etdilər [4, s.270].

Bu məqalədə araşdırılan platforma yalnız müstəqil sətir modelindən istifadə edir. Hər bir viktorina təqdimatı cəhdlər cədvəlində bir yeni sətir (cəhdlər) və cavablar cədvəlində hər sual üçün bir sətir (cavablar) yaradır. Mövcud sətirlər dəyişdirilmir. Bu seçim hər cəhd üçün analitik paneli mümkün edir; bu həm də

yaddaşın cəlb olunma ilə artdığı anlamına gəlir - bu xüsusiyyət, viktorinaları dəfələrlə sınaq tələbələr məhz ən analitik cəhətdən dəyərli məlumatları yaradanlar olduqları üçün müsbət şəkildə miqyaslanır.

Sxem strukturu: cəhdlər və cavablar cədvəlləri. Cəhdlər cədvəli (cəhdlər) hər təqdimat üçün bir sətir saxlayır. Onun sütunları bunlardır: `id` (əsas açar, avtomatik artım), `student_id` (istifadəçilər cədvəlinin xarici açarı), `quiz_id` (testlər cədvəlinin xarici açarı), `total_score` (tam ədəd, qazanılan xallar), `max_score` (tam ədəd, mümkün olan maksimum), `ötürülən` (boolean, təqdimat zamanı `total_score`-u testin keçid həddi ilə müqayisə etməklə hesablanır), `time_taken_seconds` (tam ədəd, testin başlanğıcından təqdimatına qədər olan müddət) və `submitted_at` (vaxt möhürü, verilənlər bazası tərəfindən əlavədə təyin olunur). Keçirilən sütun qəsdən artıq hesablanmış bir dəyərdir — onun dəyəri həmişə `total_score` və testin `passing_score` parametrindən əldə edilə bilər — lakin onu saxlamaq, keçidləri və uğursuzluqları saymaq üçün lazım olan hər analitik sorğuda birləşmənin qarşısını alır [7, s.42].

Cavablar cədvəli (cavablar) hər təqdimat üçün hər sual üçün bir sətir saxlayır. Onun sütunları bunlardır: `id`, `attempt_id` (cəhdlər cədvəlinin xarici açarı), `question_id` (suallar cədvəlinin xarici açarı), `selected_option` (`CHAR(1)`, nullable — `NULL`, taymerin müddəti bitdikdə cavabsız qalan sualı təmsil edir) və `is_correct` (boolean, `selected_option`-u sualın `correct_option` sahəsi ilə müqayisə etməklə təqdim etmə vaxtında hesablanır). Cəhdlərdə `ötürülən` sütunda olduğu kimi, `is_correct`-i saxlamaq, hər analitik sorğuda düzgün və səhv sayları birləşdirən suallar cədvəli vasitəsilə birləşməni artıq şəkildə aradan qaldırır [8, s.156].

Təqdim etmə əməliyyatı hər iki cədvələ atomik şəkildə yazır. Cavablar daxil etmələri cəhd sətiri təyin edildikdən sonra uğursuz olarsa, analitik panel heç bir pozuntu olmayan bir cəhd göstərəcək — pozulmuş vəziyyət. Hər iki əməliyyatı açıq `BEGIN / COMMIT` əməliyyatına sarmaq, cəhd sətirinin yalnız bütün cavab sətirləri uğurla yazıldığı təqdirdə qalmasını təmin edir. Bu, standart əməliyyat



gigiyenasıdır, lakin açıq şəkildə qeyd etməyə dəyər: analitikanın düzgünlüyü yazı atomluğundan asılıdır [9, s.201].

Analitik sorğular üçün indeks dizaynı.

Analitika panelinin əsas sorğusu üç cədvəl — cəhdlər, istifadəçilər və cavablar — birləşdirir və hər təqdimat üçün bir nəticə sətiri yaratmaq üçün cəhd ID-sinə görə qruplaşdırır. Müvafiq indekslər olmadan, bu sorğu hər təlimatçı səhifəsi yüklənməsi üçün cəhdlər cədvəlinin tam cədvəl skanına çevrilir — on tələbə ilə qəbul edilir, on min tələbə ilə istifadə edilə bilməz.

Cəhdlər cədvəlindəki iki kompozit indeks fərqli oxuma nümunələrinə xidmət edir. Birincisi, (student_id, quiz_id) üzərindəki idx_student_quiz, ən çox yayılmış analitik giriş nümunəsini gücləndirən indeksdir: "Y testində tələbə X-in bütün cəhdlərini tapın". Hər iki sütun analitik sorğunun WHERE bəndində görünür; MySQL, cədvəl məlumat sətirlərinə toxunmadan süzgəci tamamilə indeksdən həll etmək üçün kompozit indeksdən istifadə edir. İkincisi, (quiz_id, submitted_at) üzərindəki idx_quiz_submitted, fərqli bir sorğuya xidmət edir: "Y testində bütün cəhdləri zamana görə sıralayaraq tapın" — təlimatçı test pəncərəsi boyunca təqdimat fəaliyyətini görmək istədikdə faydalıdır [10, s.178].

Cavablar cədvəlində tək bir indeks var, idx_attempt on (attempt_id). Hər sual üzrə təhlil tələb edən hər bir analitik sorğu cavabları attempt_id ilə filtrləyir; bu indeks olmadan, tək bir cəhd üçün cavabları əldə etmək üçün bütün cavablar cədvəlini skan etmək lazımdır. Bununla, axtarış, viktorinadakı sualların sayı ilə məhdudlaşdırılmış bir indeks diapazonu skanıdır - adətən beşdən iyirmiyə qədər sətir [10, s.182].

Əsas analitik sorğusunda doldurulmuş verilənlər bazasına qarşı EXPLAIN işlətmək gözlənilən icra planını təsdiqləyir: cəhdlər cədvəlinə giriş üçün idx_student_quiz üzərində type=ref və cavablar cədvəlinə giriş üçün idx_attempt üzərində type=ref. Heç bir cədvəl tam skan tələb etmir. Yoxlanılan təxmini sətir sayı, hər iki cədvəlün ümumi sətir sayı ilə deyil, müəyyən bir viktorina üçün cavabların sayı ilə mütənəsibdir. Bu xüsusiyyət o deməkdir ki, sorğu vaxtı,

platformanın ümumi məlumat həcmi ilə deyil, viktorina əhatə dairəsi ilə məhdudlaşır, bu da səhifə başına yüklənmə analitikasını mümkün edən miqyaslanma xüsusiyyətidir [10, s.184].

Əsas analitik sorğusu. Təlimatçının analitik paneli, cəhdlər, istifadəçilər və cavab cədvəllərini və qruplarını cəhd ID-sinə görə birləşdirən tək bir SQL sorğusu ilə təchiz edilmişdir. Sorğu, istifadəçilər cədvəlindən tələbənin adını və elektron poçtunu, cəhd ID-sini, ümumi balı, maksimum bal, keçilmə statusunu və cəhdlər cədvəlindən alınan vaxtı, cavablar cədvəlindən isə ümumi sual sayını (COUNT), düzgün cavab sayını (is_dogru cavabın cəmi) və törəmə səhv sayını (COUNT minus SUM) seçir.

GROUP BY cəhd_id bəndi, hər cəhd üçün sətir formatını yaradan struktur qərardır. Bunun əvəzinə tələbə_id-si ilə qruplaşdırmaq, tələbənin bütün cəhdlərini tək bir aqreqat sətirə yığacaq və trayektoriya məlumatlarını itirəcək. Cəhd_id-si ilə qruplaşdırmaq hər təqdimatı ayrı bir sətir kimi saxlayır: üç cəhdi olan tələbə hər birinin öz balı və düzgün/səhv bölgüsü olan üç sətir kimi görünür. Bu, analitik panelin cədvəl komponentinin tətbiq səviyyəsində yenidən formalaşdırılmadan birbaşa istifadə etdiyi formatdır [6, s.610].

SUM(is_correct) funksiyası işləyir, çünki MySQL BOOLEAN-ı daxildə TINYINT(1) kimi saxlayır — doğru üçün 1, yalan üçün 0 — buna görə də sütunu cəmləmək düzgün cavabların sayını tam ədəd kimi verir. COUNT(id) - SUM(is_correct) funksiyası sonra alt sorğu və ya məlumatlar üzərində ikinci keçid olmadan səhv sayı verir. Bu model bəzi tətbiqlərdə görünən şərti aqreqasiyadan (SUM(CASE WHEN is_correct = 1 THEN 1 ELSE 0 END)) yayınır və bir qədər sadə sintaksislə eyni nəticələr verir [9, s.88].

WHERE bəndi, təlimatçının müəyyən bir test üçün analitik səhifəni açarkən təqdim etdiyi parametr olan quiz_id ilə filtrləyir. ORDER BY student_id, submitted_at bəndi eyni tələbənin birdən çox cəhdinin ardıcıl olaraq, xronoloji olaraq sıralanaraq görünməsinin təminatıdır — trayektoriya təhlili üçün təbii oxuma sırası. Tam sorğu parametrləşdirilir; quiz_id dəyəri icra zamanı bağlıdır və təlimatçının girişinin sorğu qatına necə

çatmasından asılı olmayaraq SQL inyeksiyasının qarşısını alır.

Nəticə

Təlim analitik ədəbiyyatında müəllim qərar qəbul etmə üçün ən faydalı hesab edilən hər bir cəhd üçün test analitikası qabiliyyəti əsasən tətbiq mürəkkəbliyi məsələsi deyil - bu, verilənlər bazası sxeminin dizaynı məsələsidir. Tələbənin əvvəlki nəticəsinin üzərinə yazmaq əvəzinə, hər bir test təqdimatını müstəqil bir sətir kimi saxlamaq qərarı sxem təyin etmə vaxtında bir dəfə verilir və tarixi məlumat itkisi olmadan geriye doğru düzəldilə bilməz.

Bu məqalədə təqdim olunan sxem, cəhdlər cədvəlindəki kompozit indekslər və GROUP BY cəhd_id analitik sorğusu birlikdə göstərir ki, tələbə başına, hər bir cəhd üçün tam bölgü - ballar, düzgün və səhv saylar, alınan vaxt, keç/uğursuzluq statusu - tam cədvəl oxunuşları əvəzinə indeks skanlarından istifadə edən tək bir SQL ifadəsində bərpa edilə bilər. Bu yanaşma standart InnoDB semantikasi olan istənilən əlaqəli verilənlər bazası sistemində tətbiq oluna bilər və xarici analitika infrastrukturunu tələb etmir.

Nəticələr göstərir ki, pulsuz LMS platformaları və kommersiya alternativləri arasındakı analitika boşluğu daxili texniki mürəkkəblik boşluğu deyil - bu, əsasən sxem dizayn boşluğudur. Kommersiya LMS lisenziyasını ala bilməyən və müəssisə yerləşdirmələri üçün sistem idarəetmə qabiliyyətinə malik olmayan qurumlar, standart açıq mənbəli komponentlər üzərində qurulmuş yüngül, davamlı tətbiqdə diqqətli sxem dizaynı vasitəsilə hər cəhd üçün ekvivalent analitik imkan əldə edə bilərlər.

ƏDƏBİYYAT SİYAHISI

1. Turnbull D., Bright L., Pahi M. Comparison of LMS platforms: moodle, canvas, and blackboard in higher education. *Journal of Educational Technology Systems*, 2019;47(3):373–394.
2. Watson W.R., Watson S.L. An argument for clarity: What are learning management systems, what are they not, and what should they become? *TechTrends*, 2007;51(2):28–34.
3. Siemens G., Long P. Penetrating the fog: Analytics in learning and education. *Educause Review*, 2011;46(5):30–32.
4. Arnold K.E., Pistilli M.D. Course signals at Purdue: Using learning analytics to increase student success. *Proceedings of LAK '12*, 2012.
5. Tanenbaum A.S., Van Steen M. *Distributed Systems: Principles and Paradigms* (3rd ed.). Pearson. 2016.
6. Romero C., Ventura S. Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics*, 2010;40(6):601–618.
7. Silberschatz A., Korth H.F., Sudarshan S. *Database System Concepts* (7th ed.). McGraw-Hill Education. 2019.
8. Connolly T., Begg C. *Database Systems: A Practical Approach to Design, Implementation, and Management* (6th ed.). Pearson. 2014.
9. MySQL AB. *MySQL 8.0 Reference Manual: InnoDB Storage Engine*. Oracle. 2023. <https://dev.mysql.com/doc/refman/8.0/en/>
10. Elmasri R., Navathe S.B. *Fundamentals of Database Systems* (7th ed.). Pearson. 2016.

Bakhish Tahir JAFAROV

Master's student at Western Caspian University, Baku, Azerbaijan

**TEST ANALYTICS ON EVERY ATTEMPT IN LEARNING MANAGEMENT SYSTEMS:
DATABASE SCHEMA DESIGN AND QUERY OPTIMIZATION**

Summary



This article examines the database design decisions that determine whether learning management systems (LMSs) can provide meaningful test analytics to teachers. A study of widely used platforms—Moodle, Canvas, Google Classroom—shows that storing data for each attempt is either nonexistent, configuration-dependent, or is behind a commercial license. The article argues that storing each test submission as a separate database row—rather than overwriting the previous score—is the most important schema decision for analytics capabilities. A federated index strategy is presented that returns each student’s performance for each attempt—score, number of correct/incorrect answers, and pass/fail—in a single SQL query without additional application logic.

Keywords: learning analytics, database schema design, relational database, SQL indexing, LMS.

Бахыш Тахир ДЖАФАРОВ

Магистрант Западно-Каспийского Университета, Баку, Азербайджан

ПОПОЛЫТАЧНАЯ АНАЛИТИКА ТЕСТИРОВАНИЯ В СИСТЕМАХ УПРАВЛЕНИЯ ОБУЧЕНИЕМ: ПРОЕКТИРОВАНИЕ СХЕМЫ БАЗЫ ДАННЫХ И ОПТИМИЗАЦИЯ ЗАПРОСОВ

Резюме

В данной статье исследуются решения по проектированию баз данных, определяющие, способна ли система управления обучением (LMS) предоставлять преподавателям значимую аналитику по каждой попытке тестирования. Обзор широко используемых платформ — Moodle, Canvas, Google Classroom — показывает, что хранение данных по каждой попытке либо отсутствует, либо зависит от конфигурации, либо скрыто за коммерческой лицензией. В статье обосновывается, что сохранение каждой отправки теста в виде отдельной строки базы данных — вместо перезаписи предыдущего результата — является наиболее значимым решением схемы для аналитических возможностей. Представлена стратегия составных индексов, позволяющая единым SQL-запросом возвращать показатели каждого студента по каждой попытке без дополнительной логики на уровне приложения.

Ключевые слова: аналитика обучения, проектирование схемы базы данных, реляционная база данных, индексирование SQL, LMS.

Daxil olub: 14.05.2026