Database Management System

Semester-III (Batch-2024)

Learning System Analytics

CHITKARA
UNIVERSITY

Supervised By:

Dr. Yogesh

Submitted By:

Aayushman Semwal 2410990244
Gautam Bhatia 2410990286
Gouravpreet Singh 2410990290
Vihaan Kashyap 2410990320

1. Table Of Content

1.Introduction

The **Learning System Analytics** is a database-driven solution designed to manage and analyse academic information efficiently. The system connects students, faculty, and administrators by providing structured records and meaningful insights into teaching and learning activities. It ensures accuracy, transparency, and better decision-making in an educational environment.

This project focuses on maintaining student profiles, course details, faculty assignments, and performance records. By applying systematic database design, normalization, and schema development, the system reduces redundancy, supports efficient data retrieval, and provides a foundation for academic analytics.

Objectives of the System:

- Maintain structured and consistent records of students, faculty, and courses.
- Provide detailed tracking of assessments and performance outcomes.
- Ensure accurate data storage and retrieval using normalization.
- Enable administrators to generate reports for academic monitoring.
- Support data-driven decisions to enhance learning and teaching.

1.1 Rules

The system follows academic rules to ensure integrity and accuracy of information. Each student and faculty member is assigned a unique ID. Course registrations are validated against prerequisites. Assessment results can only be entered by authorized faculty members. Grade modifications require administrative approval, ensuring fairness and transparency.

1.2 Procedure

The process begins with registration of students, faculty, and courses by the administration. Students enroll in courses, after which faculty manage teaching activities and record assessments. Data is stored securely in the database and can be retrieved for reports and analysis. Every update or modification is logged to ensure traceability and accountability.

1.3 Coordination

The system enables smooth coordination between students, faculty, and administrators. Faculty manage courses and assessments, students access their academic progress, and administrators oversee enrollments, course allocations, and performance tracking. The manager supervises all processes to ensure compliance with institutional policies.

1.4 Registers

Different registers are maintained to keep systematic records. The **Student Register** contains details of students and their enrollments. The **Faculty Register** records teaching responsibilities. The **Course Register** includes course details and schedules. The **Assessment Register** stores student results and grades. These registers ensure accuracy, easy monitoring, and reliable data storage.

1.5 Activities

The system involves various activities such as student enrollment, course allocation, recording of assessments, generation of reports, and analysis of performance. Additional activities include quality checks, database audits, and preparation of performance summaries for faculty and administrators. In the future, predictive analytics and early student performance monitoring can also be integrated.

2. E-R Diagram

2.1 Entities and Attributes

1. **Users**
   - Attributes: User_id (PK), Name, Email, Password_hash, Profile_image, Role
   - Represents all individuals using the system (students, faculty, or administrators).
2. **Courses**
   - Attributes: Course_id (PK), Title, Description, Category, Price, Duration_hours, Difficulty_level, Is_published, Instructor_id (FK)
   - Represents the courses offered within the system.
3. **Lessons**
   - Attributes: Lesson_id (PK), Course_id (FK), Title, Content, Video_id, Video_type, Order_index, Duration
   - Represents individual learning units or lessons within a course.
4. **Enrollments**
   - Attributes: Enrollment_id (PK), User_id (FK), Course_id (FK), Enroll_date, Progress_percentage, Time_spent, Last_viewed_lesson_id (FK)
   - Represents the record of a user enrolling in a course.
5. **Lesson_Progress**
   - Attributes: Progress_id (PK), User_id (FK), Lesson_id (FK), Completion_percentage, Watch_time, Status, Watch_count
   - Tracks student engagement and completion at the lesson level.
6. **User_Activities**
   - Attributes: Activity_id (PK), User_id (FK), Action_type, Description, Action_time
   - Represents specific user actions such as logins, submissions, or interactions.
7. **User_Engagement**
   - Attributes: Engagement_id (PK), User_id (FK), Device_type, Session_duration, Browser, App_used
   - Tracks how users engage with the system across sessions and devices.

8. **Daily_Stats**
   - Attributes: Stats_id (PK), Date, Total_logins, New_registrations, Active_users, Avg_session_duration
   - Summarizes daily activity of the system for analytics purposes.
9. **Page_Views**
   - Attributes: View_id (PK), User_id (FK), Page_url, Video_id (FK), Duration_watched, Timestamp
   - Tracks which pages and videos are accessed by users.
10. **Video_Analytics**
    - Attributes: VideoAnalytics_id (PK), User_id (FK), Course_id (FK), Event_type, Listening_time
    - Tracks specific analytics for video interactions within lessons.
11. **Course_Analytics**
    - Attributes: CourseAnalytics_id (PK), Course_id (FK), Time_spent, Progress_percentage, Completion_rate
    - Summarizes overall course-level analytics.
12. **Conversion_Funnel**
    - Attributes: Funnel_id (PK), User_id (FK), Step_name, Step_order, Completed, Time_spent
    - Represents how users progress through registration, enrollment, or purchase steps.
13. **SQL_Logs**
    - Attributes: Log_id (PK), User_id (FK), SQL_command, Execution_time, Status, Error_message, Timestamp
    - Captures system-level SQL operations for debugging and monitoring.
14. **Real_Time_Events**
    - Attributes: Event_id (PK), User_id (FK), Event_type, Event_time, Processed
    - Records user/system events happening in real time.
15. **Performance_Metrics**
    - Attributes: Metric_id (PK), Metric_name, Metric_value, Category
    - Stores general system and performance metrics for monitoring efficiency.

2.2 Relationships

1. **Enrolls In** – Users enroll in multiple courses, and each course can have multiple users. *(Users ↔ Enrollments ↔ Courses)*
2. **Contains** – Each course contains multiple lessons, and a lesson belongs to only one course.
(Courses ↔ Lessons)
3. **Tracks** – Lesson progress is tracked per user at the lesson level. *(Users ↔ Lesson_Progress ↔ Lessons)*
4. **Engages** – Users generate activities and engagement logs. *(Users ↔ User_Activities, User_Engagement)*
5. **Monitors** – Performance metrics and real-time events monitor overall system health and activity.
(Users ↔ Real_Time_Events ↔ Performance_Metrics)
6. **Analyzes** – Courses, lessons, and videos are analyzed through specialized analytics entities.
(Courses ↔ Course_Analytics, Lessons ↔ Video_Analytics, Users ↔ Conversion_Funnel, SQL_Logs)
7. **Summarizes** – Daily stats summarize user logins, registrations, and active sessions. *(Users ↔ Daily_Stats)*

2.3 Summary

- **Users** interact with **Courses** and **Lessons**, while their progress is tracked in **Lesson_Progress** and **User_Activities**.
- **Enrollments** formalize the relationship between users and courses.
- **Analytics entities** (Course_Analytics, Video_Analytics, Page_Views, Conversion_Funnel) provide detailed monitoring of user behavior and learning effectiveness.
- **System-level tracking** (SQL_Logs, Real_Time_Events, Performance_Metrics, Daily_Stats) ensures transparency, monitoring, and overall system performance.
- Together, the diagram presents a **comprehensive model** of how learning data is stored, tracked, and analyzed for decision-making.

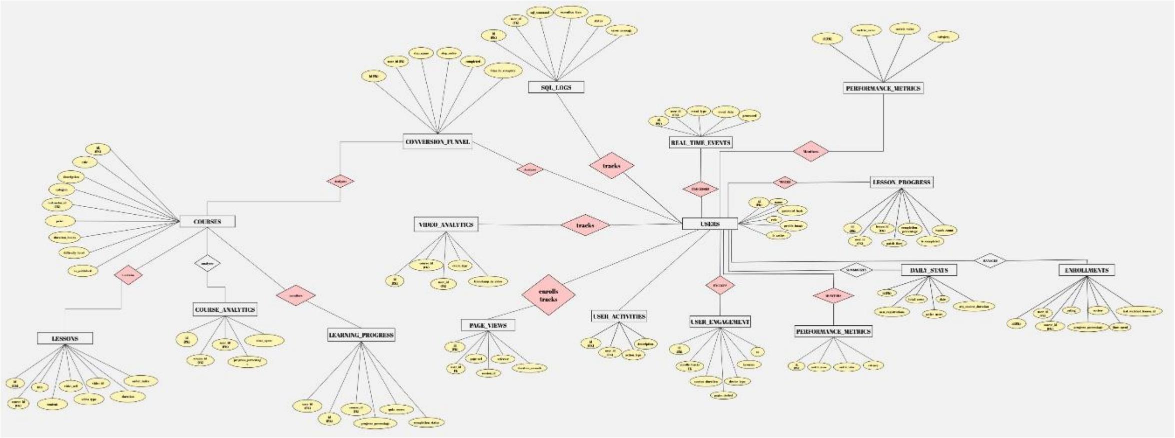


Figure 1: E-R Diagram of Learning System Analytics

https://miro.com/welcomeonboard/MytrNkRQTXluOS80NUxEdFB0R2VLVkdtMXIzdFNzMFV6NUxRZ2ZoK3N0K0wwdDZMUE40elBpNUdDbjZTTzc5ak1FL0RmNG9KaUZNTCtMRTM1YVdOb0wvcGM2dlhGK090QXNlc2ZJNmpKMzJSZVlEYlZWcEtrWG4yZHBrMEdUS1JBd044SHFHaVlWYWk0d3NxeHNmeG9BPT0hdjE=?share_link_id=934600802290

3. Normalization

Normalization is the process of organizing data to eliminate redundancy, maintain consistency, and prevent anomalies during insert, update, and delete operations. A properly normalized schema ensures that attributes depend only on the key, the whole key, and nothing but the key. The **Learning System Analytics** database adheres to these principles and is normalized up to the **Third Normal Form (3NF)**. The application of each normal form is outlined below with references to representative tables.

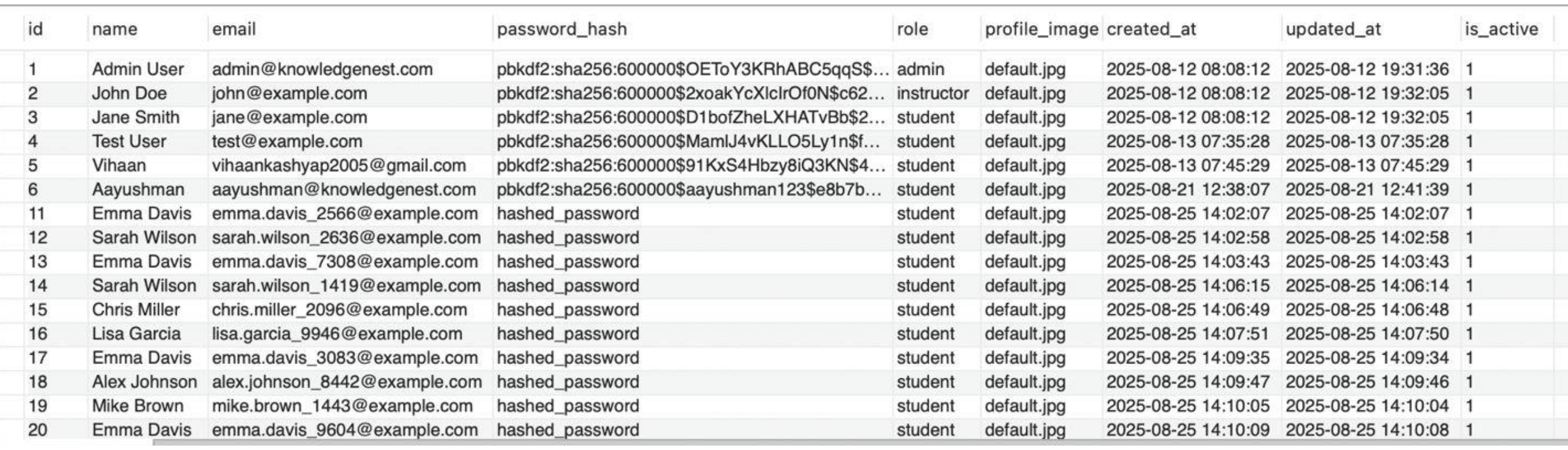
3.1 First Normal Form (1NF)

Definition. 1NF requires that every attribute holds **atomic (indivisible)** values, there are **no repeating groups or arrays** in a column, and each row represents a unique record.

Application.
In our system, operational tables are designed with atomic attributes.

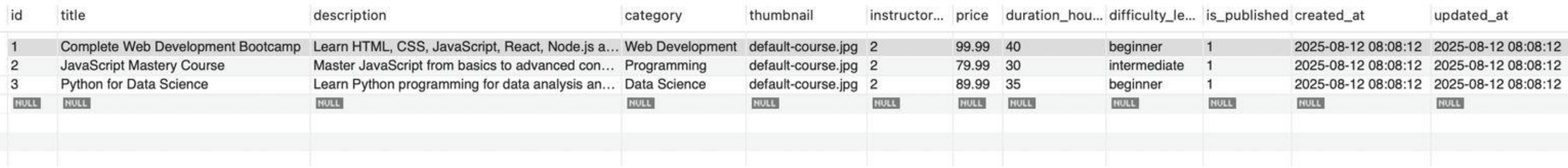
- The **Users** table stores one value per field—e.g., *name*, *email*, *role*, and *is_active* are indivisible for each user. No column contains lists or comma-separated data.
- The **Courses** table likewise maintains single values in attributes such as *title*, *description*, *category*, *price*, *duration_hours*, *difficulty_level*, and *is_published*. Each row corresponds to exactly one course record.

These choices avoid multi-valued fields, simplify querying, and support consistent indexing and validation.



id	name	email	password_hash	role	profile_image	created_at	updated_at	is_active
1	Admin User	admin@knowledgenest.com	pbkdf2:sha256:600000$OEToY3KRhABC5qqS$…	admin	default.jpg	2025-08-12 08:08:12	2025-08-12 19:31:36	1
2	John Doe	john@example.com	pbkdf2:sha256:600000$2xoakYcXlclrOf0N$c62…	instructor	default.jpg	2025-08-12 08:08:12	2025-08-12 19:32:05	1
3	Jane Smith	jane@example.com	pbkdf2:sha256:600000$D1bofZheLXHATvBb$2…	student	default.jpg	2025-08-12 08:08:12	2025-08-12 19:32:05	1
4	Test User	test@example.com	pbkdf2:sha256:600000$MamlJ4vKLLO5Ly1n$f…	student	default.jpg	2025-08-13 07:35:28	2025-08-13 07:35:28	1
5	Vihaan	vihaankashyap2005@gmail.com	pbkdf2:sha256:600000$91KxS4Hbzy8iQ3KN$4…	student	default.jpg	2025-08-13 07:45:29	2025-08-13 07:45:29	1
6	Aayushman	aayushman@knowledgenest.com	pbkdf2:sha256:600000$aayushman123$e8b7b…	student	default.jpg	2025-08-21 12:38:07	2025-08-21 12:41:39	1
11	Emma Davis	emma.davis_2566@example.com	hashed_password	student	default.jpg	2025-08-25 14:02:07	2025-08-25 14:02:07	1
12	Sarah Wilson	sarah.wilson_2636@example.com	hashed_password	student	default.jpg	2025-08-25 14:02:58	2025-08-25 14:02:58	1
13	Emma Davis	emma.davis_7308@example.com	hashed_password	student	default.jpg	2025-08-25 14:03:43	2025-08-25 14:03:43	1
14	Sarah Wilson	sarah.wilson_1419@example.com	hashed_password	student	default.jpg	2025-08-25 14:06:15	2025-08-25 14:06:14	1
15	Chris Miller	chris.miller_2096@example.com	hashed_password	student	default.jpg	2025-08-25 14:06:49	2025-08-25 14:06:48	1
16	Lisa Garcia	lisa.garcia_9946@example.com	hashed_password	student	default.jpg	2025-08-25 14:07:51	2025-08-25 14:07:50	1
17	Emma Davis	emma.davis_3083@example.com	hashed_password	student	default.jpg	2025-08-25 14:09:35	2025-08-25 14:09:34	1
18	Alex Johnson	alex.johnson_8442@example.com	hashed_password	student	default.jpg	2025-08-25 14:09:47	2025-08-25 14:09:46	1
19	Mike Brown	mike.brown_1443@example.com	hashed_password	student	default.jpg	2025-08-25 14:10:05	2025-08-25 14:10:04	1
20	Emma Davis	emma.davis_9604@example.com	hashed_password	student	default.jpg	2025-08-25 14:10:09	2025-08-25 14:10:08	1

Fig 3.1 Users Table



id	title	description	category	thumbnail	instructor...	price	duration_hou...	difficulty_le...	is_published	created_at	updated_at
1	Complete Web Development Bootcamp	Learn HTML, CSS, JavaScript, React, Node.js a...	Web Development	default-course.jpg	2	99.99	40	beginner	1	2025-08-12 08:08:12	2025-08-12 08:08:12
2	JavaScript Mastery Course	Master JavaScript from basics to advanced con...	Programming	default-course.jpg	2	79.99	30	intermediate	1	2025-08-12 08:08:12	2025-08-12 08:08:12
3	Python for Data Science	Learn Python programming for data analysis an...	Data Science	default-course.jpg	2	89.99	35	beginner	1	2025-08-12 08:08:12	2025-08-12 08:08:12
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig 3.2 Courses Table

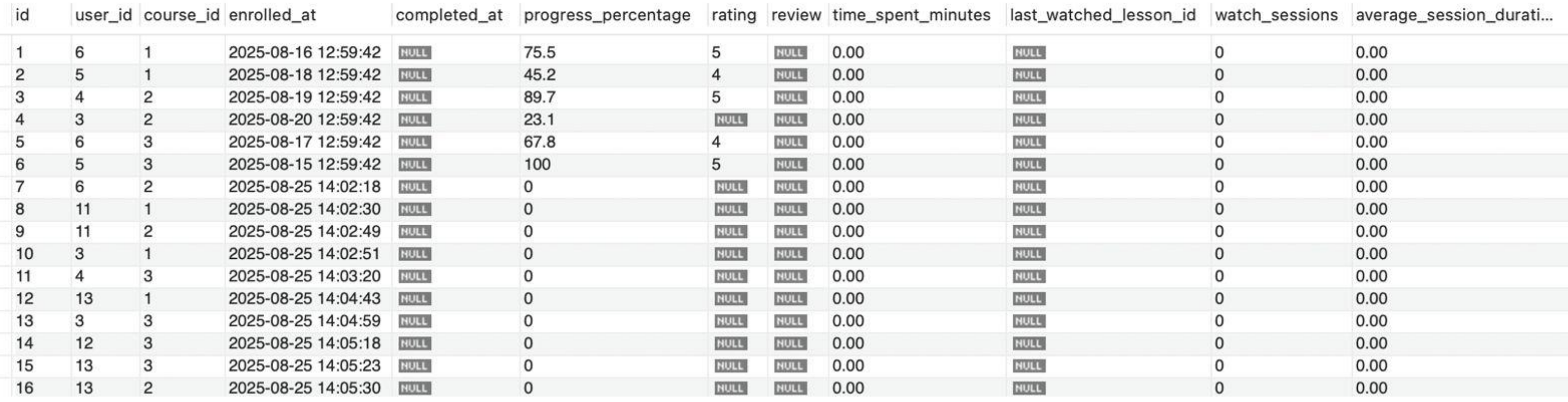
3.2 Second Normal Form (2NF)

Definition. 2NF requires that a relation already be in 1NF and that **all non-key attributes depend fully on the primary key**—i.e., no **partial dependencies** on part of a composite key.

Application.
Our core tables use **single-column primary keys** (e.g., *id*), so partial dependencies cannot arise.

- In the **Enrollments** table, the primary key is *id*. All non-key attributes—*progress_percentage*, *rating*, *review*, *time_spent_minutes*, and *last_watched_lesson_id*—depend entirely on this key, not on *user_id* or *course_id* in isolation.
- Similarly, in **Lessons** and **Lesson_Progress**, attributes depend fully on their respective single-column keys.

This ensures each non-key attribute is functionally determined by the whole key, eliminating redundancy associated with partial dependence.



id	user_id	course_id	enrolled_at	completed_at	progress_percentage	rating	review	time_spent_minutes	last_watched_lesson_id	watch_sessions	average_session_durati...
1	6	1	2025-08-16 12:59:42	NULL	75.5	5	NULL	0.00	NULL	0	0.00
2	5	1	2025-08-18 12:59:42	NULL	45.2	4	NULL	0.00	NULL	0	0.00
3	4	2	2025-08-19 12:59:42	NULL	89.7	5	NULL	0.00	NULL	0	0.00
4	3	2	2025-08-20 12:59:42	NULL	23.1	NULL	NULL	0.00	NULL	0	0.00
5	6	3	2025-08-17 12:59:42	NULL	67.8	4	NULL	0.00	NULL	0	0.00
6	5	3	2025-08-15 12:59:42	NULL	100	5	NULL	0.00	NULL	0	0.00
7	6	2	2025-08-25 14:02:18	NULL	0	NULL	NULL	0.00	NULL	0	0.00
8	11	1	2025-08-25 14:02:30	NULL	0	NULL	NULL	0.00	NULL	0	0.00
9	11	2	2025-08-25 14:02:49	NULL	0	NULL	NULL	0.00	NULL	0	0.00
10	3	1	2025-08-25 14:02:51	NULL	0	NULL	NULL	0.00	NULL	0	0.00
11	4	3	2025-08-25 14:03:20	NULL	0	NULL	NULL	0.00	NULL	0	0.00
12	13	1	2025-08-25 14:04:43	NULL	0	NULL	NULL	0.00	NULL	0	0.00
13	3	3	2025-08-25 14:04:59	NULL	0	NULL	NULL	0.00	NULL	0	0.00
14	12	3	2025-08-25 14:05:18	NULL	0	NULL	NULL	0.00	NULL	0	0.00
15	13	3	2025-08-25 14:05:23	NULL	0	NULL	NULL	0.00	NULL	0	0.00
16	13	2	2025-08-25 14:05:30	NULL	0	NULL	NULL	0.00	NULL	0	0.00

Fig 3.3 Enrollments Table

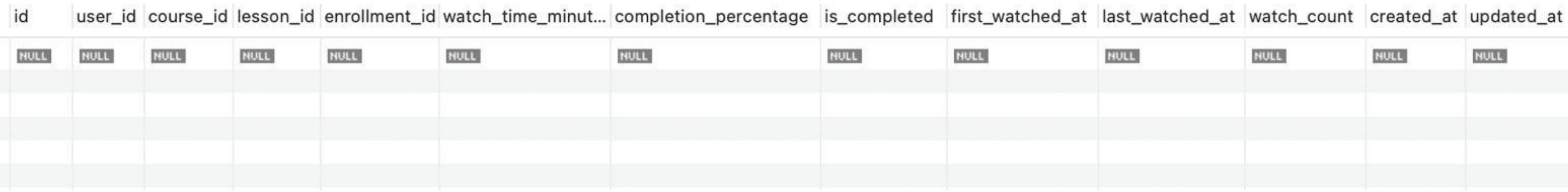
3.3 Third Normal Form (3NF)

Definition. 3NF requires that a relation be in 2NF and that **no transitive dependencies** exist—i.e., non-key attributes must **not depend on other non-key attributes**.

Application.

- In the **Lesson_Progress** table, attributes such as *watch_time_minutes*, *completion_percentage*, *is_completed*, and *watch_count* depend **only** on the primary key *id*. They do not depend on other non-key attributes.
- In **Users**, non-key attributes (*name*, *email*, *role*, *profile_image*, *is_active*) depend solely on *id* and not on one another; the same holds for **Courses**, where attributes like *price* and *difficulty_level* depend only on *id*.

Note (integrity vs. normalization): Applying a **UNIQUE** constraint to *email* in **Users** improves integrity (one account per email) but **does not change** the table's 1NF/2NF/3NF status, because uniqueness constraints don't alter functional dependencies in this design.



id	user_id	course_id	lesson_id	enrollment_id	watch_time_minut...	completion_percentage	is_completed	first_watched_at	last_watched_at	watch_count	created_at	updated_at
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig 3.4 Lesson Progress Table

3.4 Conclusion

By applying **1NF** (atomic values and no repeating groups), **2NF** (elimination of partial dependencies via single-column primary keys), and **3NF** (removal of transitive dependencies), the operational schema of the **Learning System Analytics** database is **normalized up to 3NF**. This structure minimizes redundancy, reduces update anomalies, and supports efficient, reliable query processing. Where high-volume reporting requires it, limited denormalization may be used in analytics tables; however, the core application tables demonstrated above adhere strictly to 3NF.

4. Database Schema

```
Create database knowledgenest_db;
use knowledgenest_db;

-- 1. Users Table --
CREATE TABLE Users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(120) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    role ENUM('student','instructor','admin') DEFAULT 'student',
    profile_image VARCHAR(255) DEFAULT 'default.jpg',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    is_active BOOLEAN DEFAULT TRUE
);

-- 2. Courses Table --
CREATE TABLE Courses (
    id INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(200) NOT NULL,
    description TEXT,
    category VARCHAR(100) NOT NULL,
    instructor_id INT NOT NULL,
    price DECIMAL(10,2) DEFAULT 0.00,
    duration_hours INT DEFAULT 0,
    difficulty_level ENUM('beginner','intermediate','advanced') DEFAULT 'beginner',
    is_published BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (instructor_id) REFERENCES Users(id)
);

-- 3. Lessons Table --
CREATE TABLE Lessons (
    id INT PRIMARY KEY AUTO_INCREMENT,
    course_id INT NOT NULL,
    title VARCHAR(200) NOT NULL,
    content TEXT,
    video_url VARCHAR(500),
    order_index INT DEFAULT 0,
    duration_minutes INT DEFAULT 0,
    video_type ENUM('youtube','vimeo','local','external') DEFAULT 'youtube',
    video_id VARCHAR(255),
    is_preview BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (course_id) REFERENCES Courses(id)
);

-- 4. Enrollments Table --
CREATE TABLE Enrollments (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    course_id INT NOT NULL,
    enrolled_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    completed_at TIMESTAMP NULL,
    progress_percentage FLOAT DEFAULT 0.0,
    rating INT,
    review TEXT,
    time_spent_minutes DECIMAL(10,2) DEFAULT 0.0,
    last_watched_lesson_id INT DEFAULT NULL,
    FOREIGN KEY (user_id) REFERENCES Users(id),
    FOREIGN KEY (course_id) REFERENCES Courses(id),
    UNIQUE (user_id, course_id)
);

-- =====================================================
-- TRACKING TABLES
-- =====================================================

-- 5. Lesson Progress Table --
CREATE TABLE Lesson_Progress (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    course_id INT NOT NULL,
    lesson_id INT NOT NULL,
    enrollment_id INT NOT NULL,
    watch_time_minutes DECIMAL(10,2) DEFAULT 0.0,
    completion_percentage FLOAT DEFAULT 0.0,
    is_completed BOOLEAN DEFAULT FALSE,
    watch_count INT DEFAULT 0,
    first_watched_at TIMESTAMP NULL,
    last_watched_at TIMESTAMP NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(id),
    FOREIGN KEY (course_id) REFERENCES Courses(id),
    FOREIGN KEY (lesson_id) REFERENCES Lessons(id),
    FOREIGN KEY (enrollment_id) REFERENCES Enrollments(id),
    UNIQUE (user_id, course_id, lesson_id)
);

-- 6. User Activities Table --
CREATE TABLE User_Activities (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    action_type ENUM('login','register','enrollment','completion','logout') NOT NULL,
    description VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(id)
);

-- 7. Video Analytics Table --
CREATE TABLE Video_Analytics (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    lesson_id INT NOT NULL,
    event_type ENUM('play','pause','seek','complete','exit') NOT NULL,
    timestamp_in_video INT DEFAULT 0,
    event_timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(id),
    FOREIGN KEY (lesson_id) REFERENCES Lessons(id)
);

-- 8. SQL Logs Table --
```

```
CREATE TABLE SQL_Logs (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    sql_command TEXT NOT NULL,
    command_type ENUM('SELECT','INSERT','UPDATE','DELETE','OTHER') NOT NULL,
    execution_time_ms DECIMAL(10,3),
    status ENUM('SUCCESS','ERROR','WARNING') NOT NULL,
    error_message TEXT,
    executed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(id)
);

-- ============================================================
-- ANALYTICS TABLES
-- ============================================================

-- 9. Page Views Table --
CREATE TABLE Page_Views (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    page_url VARCHAR(500) NOT NULL,
    session_id VARCHAR(100),
    duration_seconds INT DEFAULT 0,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(id)
);

-- 10. User Engagement Table --
CREATE TABLE User_Engagement (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    session_duration_minutes INT DEFAULT 0,
    pages_visited INT DEFAULT 0,
    device_type ENUM('desktop','tablet','mobile') DEFAULT 'desktop',
    browser VARCHAR(100),
    os VARCHAR(100),
    FOREIGN KEY (user_id) REFERENCES Users(id)
);

-- 11. Course Analytics Table --
CREATE TABLE Course_Analytics (
    id INT PRIMARY KEY AUTO_INCREMENT,
    course_id INT NOT NULL,
    user_id INT NOT NULL,
    event_type ENUM('view','start','progress','complete','pause','resume') NOT NULL,
    progress_percentage FLOAT DEFAULT 0.0,
    time_spent_minutes INT DEFAULT 0,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (course_id) REFERENCES Courses(id),
    FOREIGN KEY (user_id) REFERENCES Users(id)
);

-- 12. Learning Progress Table --
CREATE TABLE Learning_Progress (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    course_id INT NOT NULL,
    progress_percentage FLOAT DEFAULT 0.0,
    quiz_scores TEXT,
    completion_status VARCHAR(50),
    FOREIGN KEY (user_id) REFERENCES Users(id),
    FOREIGN KEY (course_id) REFERENCES Courses(id)
);

-- 13. Performance Metrics Table --
CREATE TABLE Performance_Metrics (
    id INT PRIMARY KEY AUTO_INCREMENT,
    metric_name VARCHAR(100) NOT NULL,
    metric_value FLOAT NOT NULL,
    category ENUM('response_time','throughput','error_rate','resource_usage') NOT NULL,
    recorded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- 14. Daily Stats Table --
CREATE TABLE Daily_Stats (
    id INT PRIMARY KEY AUTO_INCREMENT,
    date DATE NOT NULL UNIQUE,
    total_users INT DEFAULT 0,
    active_users INT DEFAULT 0,
    new_registrations INT DEFAULT 0,
    lessons_completed INT DEFAULT 0,
    avg_session_duration FLOAT DEFAULT 0.0,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- 15. Conversion Funnel Table --
CREATE TABLE Conversion_Funnel (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    step_name VARCHAR(100) NOT NULL,
    step_order INT NOT NULL,
    completed BOOLEAN DEFAULT FALSE,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(id)
);
```

5. References

[1] Romero, C., & Ventura, S., “Educational Data Mining and Learning Analytics: An Updated Survey,” *arXiv*. [Online]. Available: https://arxiv.org/abs/2402.07956. Accessed: July 12, 2025.

[2] Hernández-de-Menéndez, M., Vallejo Guevara, A., Morales-Menendez, R., & Escobar, C. A., “Learning Analytics: State of the Art,” *PMC*. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC9206225/.
Accessed: July 19, 2025.

[3] Estuary.dev, “Data Normalization Explained: Types, Examples, & Methods,” *Estuary*. [Online]. Available: https://estuary.dev/blog/data-normalization/.
Accessed: July 23, 2025.

[4] Analytics Vidhya, “Database Normalization: A Step-by-Step Guide with Examples,” *Analytics Vidhya*. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/08/database-normalization-a-step-by-step-guide-with-examples/.
Accessed: July 27, 2025.

[5] Wikipedia, “Learning Analytics,” *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Learning_analytics.
Accessed: July 14, 2025.

[6] Wikipedia, “Database Normalization,” *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Database_normalization.
Accessed: July 20, 2025.

[7] Nguyen, A., “Design Principles for Learning Analytics Information Systems,” *Learning: Research and Practice*. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/0960085X.2020.1816144.
Accessed: July 25, 2025.