

# Entity Relationship Diagram- document

## Learning Recommendation System

### Database Description:

This database contains 8 tables.

### A. Tables and Attributes with Constraints

#### 1. Users (sd\_USERS)

- **Primary Key:** `sd_user_id` (INT, PRIMARY KEY)
- **Attributes:**
  - `sd_USER_first_name` (VARCHAR(50), NOT NULL) – Stores the first name of the user.
  - `sd_USER_last_name` (VARCHAR(50), NULL) – Stores the last name (optional).
  - `sd_USER_email` (VARCHAR(60), UNIQUE, NOT NULL) – Stores a unique email ID.
  - `sd_USER_preferences` (VARCHAR(100), NULL) – Stores user preferences.
  - `sd_USER_signup_date` (DATE, NULL) – Stores the signup date.
- **Description:** Stores information about users of the platform.
- **Relationships:**
  - **One-to-many** with `sd_COURSE_OUTLINES`, `sd_COMPLETION_RECORDS`, `sd_REVIEWS`, and `sd_RECOMMENDATIONS`.

## 2. Categories (sd\_CATEGORIES)

- **Primary Key:** sd\_CATEGORY\_ID (INT, PRIMARY KEY)
- **Attributes:**
  - sd\_CATEGORY\_name (VARCHAR(50), UNIQUE, NOT NULL) – Stores the category name.
  - sd\_CATEGORY\_DESCRIPTION (TEXT, NULL) – Stores a detailed category description.
- **Description:** Represents course categories for classification purposes.
- **Relationships:**
  - **One-to-many** with sd\_COURSES.

## 3. Courses (sd\_COURSES)

- **Primary Key:** sd\_COURSE\_ID (INT, PRIMARY KEY)
- **Attributes:**
  - sd\_COURSE\_NAME (VARCHAR(255), NOT NULL) – Stores the course name.
  - sd\_CATEGORY\_ID (INT, NULL, FOREIGN KEY → sd\_CATEGORIES(sd\_CATEGORY\_ID)) – Links courses to categories.
  - sd\_COURSE\_RATING (DECIMAL(3,2), CHECK BETWEEN 0 AND 5, NULL) – Stores course rating between 0 and 5.
- **Description:** Stores details of courses available for recommendation.
- **Relationships:**
  - **Many-to-one** with **Categories**.
  - **One-to-many** with sd\_COMPLETION\_RECORDS, sd\_REVIEWS, sd\_RECOMMENDATIONS, and sd\_OUTLINE\_RECOMMENDATIONS .

## 4. Reviews (sd\_REVIEWS)

- **Primary Key:** sd\_REVIEW\_ID (INT, PRIMARY KEY)
- **Attributes:**

- `sd_USER_ID` (INT, NOT NULL, FOREIGN KEY → `sd_USERS(sd_user_id)` ON DELETE CASCADE) – Links review to a user.
- `sd_COURSE_ID` (INT, NOT NULL, FOREIGN KEY → `sd_COURSES(sd_COURSE_ID)` ON DELETE CASCADE) – Links review to a course.
- `sd_REVIEW_RATING` (INT, CHECK BETWEEN 1 AND 5, NOT NULL) – Stores rating between 1 and 5.
- `sd_REVIEW` (TEXT, NULL) – Stores the review text.
- **Description: Stores user reviews and ratings for courses.**
- **Relationships:**
  - **Many-to-one** with `sd_USERS`.
  - **Many-to-one** with `sd_COURSES`.

## 5. Completion Records (`sd_COMPLETION_RECORDS`)

- **Primary Key:** `sd_RECORD_ID` (INT, PRIMARY KEY)
- **Attributes:**
  - `sd_USER_ID` (INT, NOT NULL, FOREIGN KEY → `sd_USERS(sd_user_id)` ON DELETE CASCADE) – Links to a user.
  - `sd_COURSE_ID` (INT, NOT NULL, FOREIGN KEY → `sd_COURSES(sd_COURSE_ID)` ON DELETE CASCADE) – Links to a course.
  - `sd_COMPLETION_DATE` (DATE, NULL) – Stores course completion date.
  - `sd_PROGRESS_PERCENTAGE` (DECIMAL(5,2), CHECK BETWEEN 0 AND 100, NULL) – Progress in percentage.
- **Description: Tracks course progress or completion status for users.**
- **Relationships:**
  - **Many-to-one** with `sd_USERS`.
  - **Many-to-one** with `sd_COURSES`.

## 6. Course Outlines (`sd_COURSE_OUTLINES`)

- **Primary Key:** `sd_OUTLINE_ID` (INT, PRIMARY KEY)
- **Attributes:**

- `sd_USER_ID` (INT, NOT NULL, FOREIGN KEY → `sd_USERS(sd_user_id)` ON DELETE CASCADE) – Links outline to a user.
- `sd_COURSE_OUTLINE_CONTENT` (TEXT, NOT NULL) – Stores outline content.
- **Description:** Stores course outlines uploaded by users.
- **Relationships:**
  - **Many-to-one** with `sd_USERS`.
  - **One-to-many** with `sd_OUTLINE_RECOMMENDATIONS`.

## 7. Recommendations (`sd_RECOMMENDATIONS`)

- **Primary Key:** `sd_RECOMMENDATION_ID` (INT, PRIMARY KEY)
- **Attributes:**
  - `sd_USER_ID` (INT, NOT NULL, FOREIGN KEY → `sd_USERS(sd_user_id)` ON DELETE CASCADE) – Links recommendation to a user.
  - `sd_COURSE_ID` (INT, NOT NULL, FOREIGN KEY → `sd_COURSES(sd_COURSE_ID)` ON DELETE CASCADE) – Links recommendation to a course.
  - `sd_RECOMMENDATION_LOGIC` (TEXT, NOT NULL) – Stores recommendation logic.
- **Description:** Represents recommendations provided to users based on their preferences and history.
- **Relationships:**
  - **Many-to-one** with `sd_USERS`.
  - **Many-to-one** with `sd_COURSES`.

## 8. Outline Recommendations (`sd_OUTLINE_RECOMMENDATIONS`)

- **Primary Key:** `sd_RECOMMENDATION_ID` (INT, PRIMARY KEY, AUTO\_INCREMENT)
- **Attributes:**

- `sd_OUTLINE_ID` (INT, NOT NULL, FOREIGN KEY → `sd_COURSE_OUTLINES(sd_OUTLINE_ID)` ON DELETE CASCADE) – Links recommendation to an outline.
- `sd_COURSE_ID` (INT, NOT NULL, FOREIGN KEY → `sd_COURSES(sd_COURSE_ID)` ON DELETE CASCADE) – Links recommendation to a course.
- `sd_MATCH_SCORE` (DECIMAL(5,2), CHECK BETWEEN 0 AND 100, NULL) – Stores matching score percentage.
- **Description: Represents recommendations based on user-uploaded course outlines.**
- **Relationships:**
  - **Many-to-one with** `sd_COURSE_OUTLINES`
  - **Many-to-one with** `sd_COURSES`.

## B. Relationships in Detail

### 1. Users → Reviews (1:M)

- A **User** can write multiple **Reviews**, but each **Review** belongs to only one **User**.
- **Foreign Key:** `sd_USER_ID` in `sd_REVIEWS` references `sd_USERS.sd_user_id`.

### 2. Users → CompletionRecords (1:M)

- A **User** can complete multiple **Courses**, but each **CompletionRecord** belongs to only one **User**.
- **Foreign Key:** `sd_USER_ID` in `sd_COMPLETION_RECORDS` references `sd_USERS.sd_user_id`.

### 3. Users → Recommendations (1:M)

- A **User** can receive multiple **Course Recommendations**, but each **Recommendation** is linked to only one **User**.
- **Foreign Key:** `sd_USER_ID` in `sd_RECOMMENDATIONS` references `sd_USERS.sd_user_id`.

### 4. Courses → Reviews (1:M)

- A **Course** can have multiple **Reviews**, but each **Review** is linked to only one **Course**.

- **Foreign Key:** `sd_COURSE_ID` in `sd_REVIEWS` references `sd_COURSES.sd_COURSE_ID`.
5. **Courses → CompletionRecords (1:M)**
- A **Course** can be completed by multiple **Users**, but each **CompletionRecord** is linked to only one **Course**.
  - **Foreign Key:** `sd_COURSE_ID` in `sd_COMPLETION_RECORDS` references `sd_COURSES.sd_COURSE_ID`.
6. **Courses → Recommendations (1:M)**
- A **Course** can be recommended to multiple **Users**, but each **Recommendation** is linked to only one **Course**.
  - **Foreign Key:** `sd_COURSE_ID` in `sd_RECOMMENDATIONS` references `sd_COURSES.sd_COURSE_ID`.
7. **Categories → Courses (1:M)**
- A **Category** can have multiple **Courses**, but each **Course** belongs to only one **Category**.
  - **Foreign Key:** `sd_CATEGORY_ID` in `sd_COURSES` references `sd_CATEGORIES.sd_CATEGORY_ID`.
8. **Users → CourseOutlines (1:M)**
- A **User** can create multiple **Course Outlines**, but each **Outline** is linked to only one **User**.
  - **Foreign Key:** `sd_USER_ID` in `sd_COURSE_OUTLINES` references `sd_USERS.sd_user_id`.
9. **CourseOutlines → OutlineRecommendations (1:M)**
- A **Course Outline** can have multiple **Outline Recommendations**, but each **Recommendation** is linked to only one **Outline**.
  - **Foreign Key:** `sd_OUTLINE_ID` in `sd_OUTLINE_RECOMMENDATIONS` references `sd_COURSE_OUTLINES.sd_OUTLINE_ID`.
10. **Courses → OutlineRecommendations (1:M)**
- A **Course** can be linked to multiple **Outline Recommendations**, but each **Recommendation** is tied to only one **Course**.
  - **Foreign Key:** `sd_COURSE_ID` in `sd_OUTLINE_RECOMMENDATIONS` references `sd_COURSES.sd_COURSE_ID`.