Analysis Steps:

1- Define the Scope:

Recorded Courses System

2- define the Manditotry object:

- Courses
- Learners
- Admin

3- Software Type:

Customized

4-sequential flow:

1.Admin:

- Login to the system
- should be able to add users by email,Name, password, profile image, and bio (users=>Instructors & Learners)
- should be able to explore :
    1. users by email,Name, img
    2. Courses by title, description, Rate, and price
    3. Course details by title, description, level, price, discount , total duration , Rate , Reviews,category
- Admin should be able Active, or inactive users
- Admin should be able approve/disapprove Course publish
- Admin should be able to generate analytics reports
- Admin should be able to manage subscription plans (add, update, delete)=> free,Basic,Premium
- Admin should be able to create and manage course bundles (grouping multiple Course)

2.Instructors:

- Login to the system
- instructors should be able to add Courses by title, description, category, level, price, discount ,total duration
- Instructors should be able to Manage(Add, Update, Activation) Lesson by (video file, title, and duration)

- should be able to logout

3.Learners

- Login to the system
- should be able to browse, search, and filter courses
- should be able to Enroll in courses and track progress for each course and each lessons (1:M) =>Course Progress maybe derived Attribute (total Lesson complete /total Lesson *100%)
- should be able to Bookmark favorite courses and leave reviews for courses and instructors.
- should be able to logout

5- Summarizing System Object:

- Admin (email,Name, password)
- Instructors (email,Name, password, profile image, and bio)
- Courses (title, description, category, level, price, discount ,total duration)
- Lessons (video file, title, and duration)
- Learner (email,Name, password, profile image, and bio)
- Review (comment,rate)
- subscription plans (type subscription , Duration,price)
- course bundles (name,course include,price,des,discount,status(active or inactive))
- Bookmark CouseLearnerTable

6- Generalization and specialization (Restructuring)

- user:Admin, Instructor, Learner
- Members: instructor,Learner
- Members,Courses,lesson,Review ,subscription plans,course bundles,Bookmark : parent Entity (id,created by , created At,Update by,Update At,IsActive)
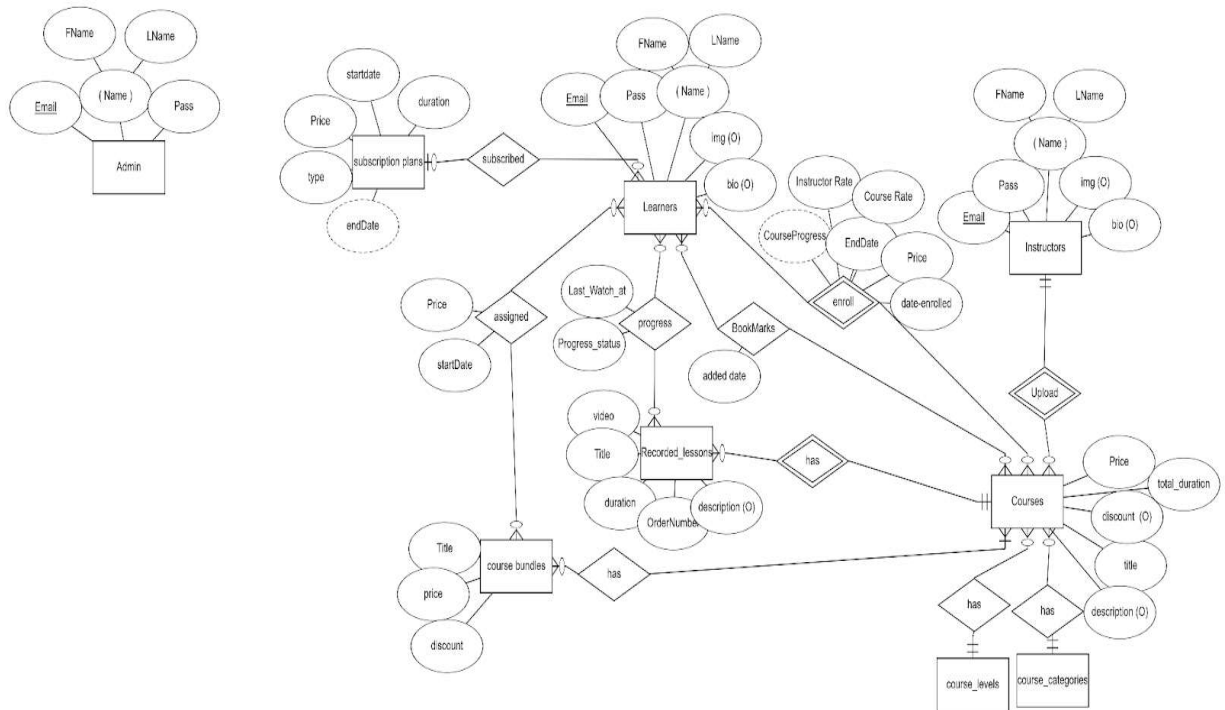
7- Define the Relationship

inheritance:

- single table inheritance
- concurrentTable inheritance: parent entity
- Class table inheritance :can use user (instructor,learner) as table inheritance

Composition:

- Instructors & Courses =>1:M
- Course & Lesson => 1:M
- Learner & Course => M:M (Enrolled Course)
- Learner & Course=>M:M (Bookmark favorite courses)
- Learner & Lesson => M:M
- Learner & Course bundles =>M:M
- Course bundles & courses=> M:M
- Learner & subscription plan

8- ER diagram

9- DataBase:

Normalization:

- Actors:
  - instructors
  - Learners
  - Admin

- object:
  - courses
  - Lesson
  - course bundles

- ○ subscription plan

- **Lookups:**
  - ○ **course category**
  - ○ **course level**
  - ○ **subscription type**
  - ○ **lesson status**

schema:

Admin
- admin_id
- Email
- password
- FName
- LName

courses_bundles
- courses_bundles_id
- is_active

bundles
- bundles_id
- discount

learner_bundles
- learner_bundles_id
- start_date

courses
- course_id
- Title
- description
- price
- discount
- CreatedAt
- UpdatedAt
- IsActive
- instructor_id
- category_id
- level_id

bookmarks
- bookmark_id
- date_added

learner_courses
- price

Learners
- learner_id
- Email
- password
- FName
- LName
- img_url
- bio

recorded_lessons
- lesson_id
- order_number

Lockups_item
- item_id
- name

learner_lesson_progress
- lesson_progress_id
- last_watch_at
- learner_id
- lesson_id
- lesson_status

Instructors
- instructors_id
- Email
- password
- FName
- LName

subscription
- sub_id
- discount

Lockups_type
- type_id
- type
- IsActive