# ■ Sequelize Syllabus (Beginner → Advanced)

## 1. Introduction & Setup

- What is Sequelize? ORM vs. raw SQL
- Installing Sequelize & DB drivers (mysql2, pg, etc.)
- Connecting to database
- Sequelize CLI (init, migrations, seeders)

## 2. Models & DataTypes

- Defining models
- DataTypes (STRING, INTEGER, BOOLEAN, DATE, JSON, etc.)
- Model options (timestamps, underscored, paranoid, freezeTableName)
- Table naming conventions (plural vs. singular)

## 3. Basic CRUD Operations

- Creating records (create, bulkCreate)
- Reading records (findOne, findAll, findByPk)
- Updating records (update, save)
- Deleting records (destroy)
- Raw SQL queries with sequelize.query

## 4. Querying & Operators

- where conditions
- Sequelize operators (Op.gt, Op.like, Op.or, etc.)
- Ordering, grouping, aggregations (count, sum, max, min)
- Pagination (limit, offset)

## 5. Associations (Relationships)

- One-to-One (hasOne, belongsTo)
- One-to-Many (hasMany, belongsTo)
- Many-to-Many (belongsToMany with through)
- Foreign keys & aliases (as)
- Eager loading with include
- Nested includes (User → Post → Comment)
- Magic methods (addX, getX, setX, removeX)

## 6. Migrations & Seeders

- Creating migrations with Sequelize CLI
- Running migrations (db:migrate)
- Rolling back (db:migrate:undo)
- Creating seeders for dummy data
- Managing schema changes in teams

## 7. Validations & Constraints

- Built-in validations (allowNull, unique, validate)
- Custom validators
- Constraints (NOT NULL, UNIQUE, CHECK)

## 8. Hooks (Lifecycle Events)

- Before / After hooks (beforeCreate, afterUpdate, etc.)
- Use cases: hashing passwords, logging changes, sending emails

## 9. Transactions

- What are transactions & why use them
- Managed transactions (sequelize.transaction)
- Unmanaged transactions (manual commit/rollback)

## 10. Scopes & Advanced Queries

- Default scopes
- Named scopes
- Complex queries with scopes
- Virtual fields (VIRTUAL)

## 11. Performance Optimization

- Eager vs. lazy loading
- Query batching
- Indexing & optimizing queries
- Using raw queries when necessary

## 12. Advanced Topics

- Polymorphic associations
- Self-associations (e.g., User → Manager User)
- Through models with extra fields (e.g., Enrollment with grade)
- Paranoid models (soft deletes)
- Hierarchical data (trees, nested sets)

## 13. Testing

- Unit testing models with Jest
- Mocking Sequelize in tests
- Using SQLite in-memory DB for testing

## 14. Integration with Express & React

- Structuring Sequelize with Express (models/, controllers/, routes/)
- Exposing Sequelize data via REST APIs

- Consuming APIs in React frontend
- Error handling & validation responses

## ■ Final Project (Capstone)

- Build a Mini HRMS or Blog App:
- - Users (with profiles) → one-to-one
- - Users & Posts → one-to-many
- - Posts & Tags → many-to-many
- - Use migrations, seeders, hooks, transactions
- - Full CRUD APIs with Express
- - React frontend to consume APIs