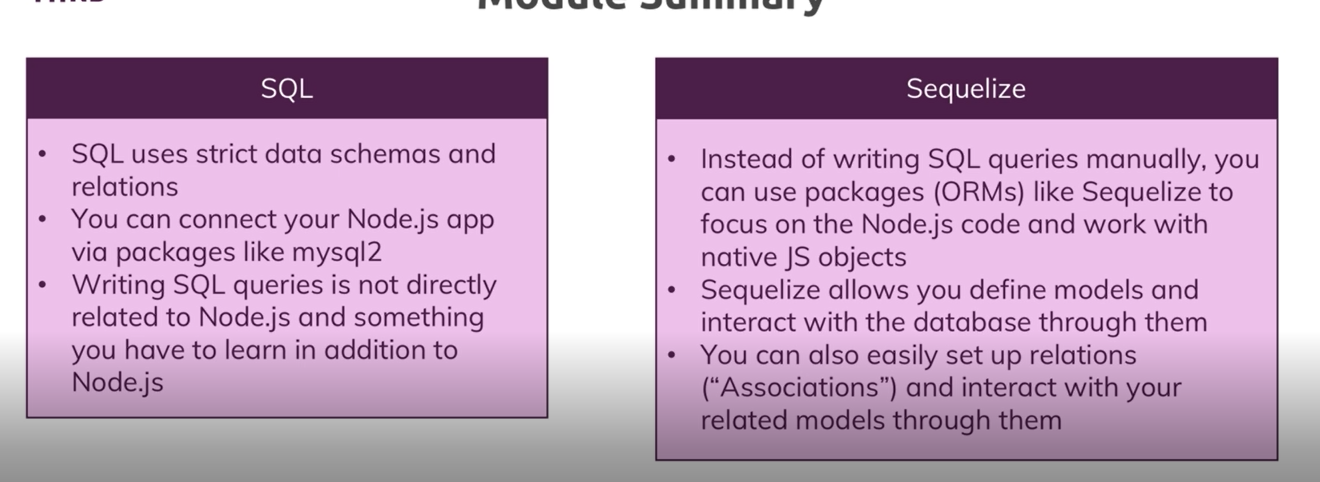
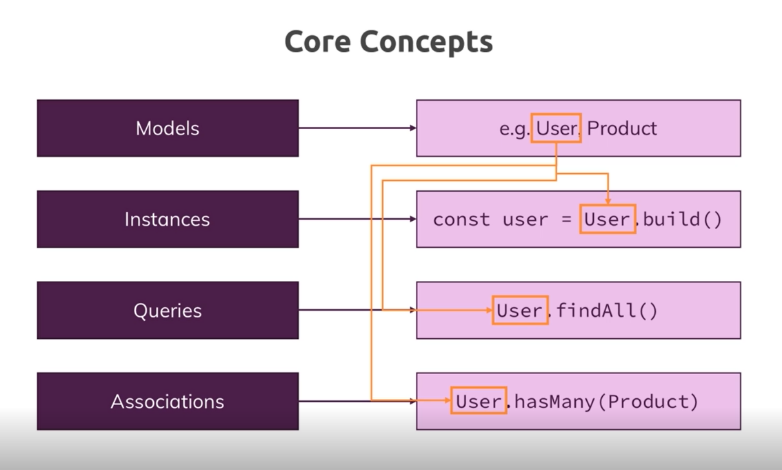
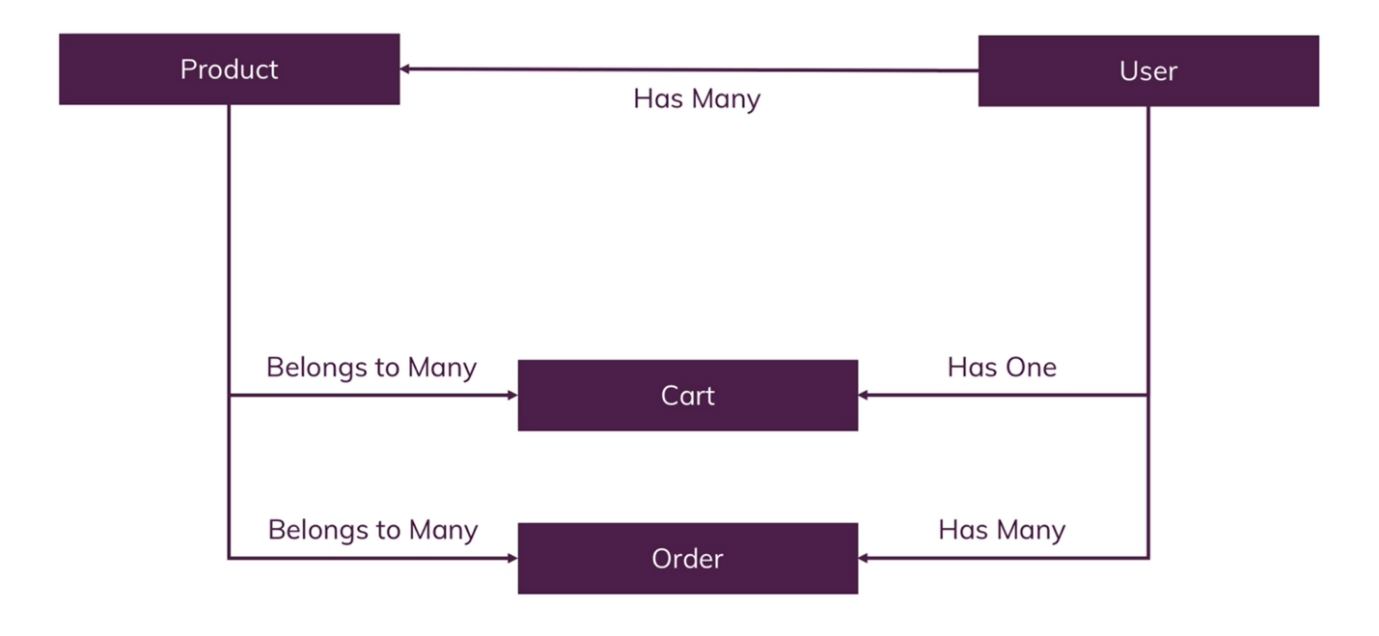
**Sequelize**[**https://sequelize.org/v3/**](https://sequelize.org/v3/)

**Summary**Due to this package, we will focus on Nodejs not SQL.  
we will use mqsql db in background but the cod we write will be different (query)

1. **what is Sequelize**  
   It is an Object Relational Mapping Library (ORM) which means it handle all the heavy lifting of SQL behind the screen and maps it to JSON object.  
   
2. **Connecting with Sequelize** a. install sequalize package. (and make sure mysql2 is also installed)  
    b. first we connect sequelize with database, by importing sequelize and creating object of it.  
   const sequelize = new Sequelize('node-complete', 'root', 'Samarjeet1!', { dialect: 'mysql', host: 'localhost' });
3. **Defining model**To define a model we import database.js file and use method of reference of sequelize object.   
   sequelize.define(param1, {}); where param1 is name of model and params is map of key (is attribute) and value (is the contrain) of that attribute.
4. **Syncing Js definition to the database**before running our app we have to make sure that tables of that sequelize model exists in database. If not we have to create that and if exists we don’t need to override it to make new table.  
     
   we can this, with the help of sync() method of sequelize reference in database.js file.  
     
   write this in app.js file at last, in then() we start our app.
5. **Inserting data & Creating a Product**to insert a add we will create() on the exported model. It takes map as a parameter where key is attribute and value is data.
6. **Retrieving data & Finding Products**a. to get all data we will findAll() on the exported model. It will return all data of table  
   b. to get conditional data we will **findAll({where: })** on the exported model and pass map of conditions. OR we can use **findByPk(id**) which takes an id as parameter and return single value while findAll return array of value
7. **Updating data**a. to update a value first we have to find that value from the table we can do that using findByPk().  
   b. then() of findByPk we will get a product and now we will override the attributes of product   
   eg. product.title = updatedTitle;  
   this we update product value locally not in database  
   c. to save updated data in database we will use product.save()
8. **Deleting data**There are 2 approaches to delete product  
     
   A. Deleting by findByPk()   
    a. first we will find that product using findByPk() and in then() we   
   will use product.destory() to delete product with pk is equal to id.  
     
   B. Deleting by destroy()  
    a. in Product.destory({where: id == id}) in this we use destroy() on imported sequelize and we pass map as a parameter where we write different condition.
9. **Associations / Relations**a. we can relate two tables using belongsTo() on models in app.js file above sync() and we can pass constraints as params  
   eg. Product.belongsTo(User)  
   b. we have inverse relation using hasMany()  
   eg. User.hasMany(Product)  
   c. above functions are used to create relations
10. **Magic Association methods**a. when we create an association sequelize create special function which make thing easy  
    eg:   
    req.user.createProduct({

title: title,

price: price,

imageUrl: imageUrl,

description: description,

});  
above function is a special function which add product for that user

Product.create({

title: title,

price: price,

imageUrl: imageUrl,

description: description,

userId: req.user.id,  
});

1. **1-to-many & many-to-many relation**a. bhai doc read kar le ek bar boot complex hai
2. **Pls read doc for relation**