const ContentVideoView = sequelize.define('ContentVideoView', {

time: {

type: DataTypes.INTEGER,

allowNull: false

}

});

ContentVideoView.associate = function (models) {

ContentVideoView.belongsTo(models.User);

ContentVideoView.belongsTo(models.CourseContent, {

foreignKey: {

name: 'coursecontentId',

allowNull: false

}

});

};

const CourseContent = sequelize.define('CourseContent', {

title: {

type: DataTypes.STRING(50),

allowNull: false

},

description: {

type: DataTypes.STRING(150),

allowNull: false

},

});

CourseContent.associate = function (models) {

CourseContent.hasMany(models.ContentVideoView, {

foreignKey: {

name: 'coursecontentId',

allowNull: false

}

});

};

WE use both hasmany many and belong relation with tables. For example course content has multiple videos.

Nested Include:

var result = await models.Course.findOne({

where: { id },

include: [{model:models.CourseContent,

include: [{model:models.ContentVideoView,separate:true,

where:{

UserId:req.user.id

}}],

separate:true,

order: [['priority', 'asc']],

}, models.Category],

});