

关联

```
// 1:N关系
const Player = sequelize.define('player', {name: Sequelize.STRING});
const Team = sequelize.define('team', {name: Sequelize.STRING});

// 会添加teamId到Player表作为外键
Player.belongsTo(Team); // 1端建立关系
Team.hasMany(Player); // N端建立关系

// 同步
sequelize.sync({force:true}).then(async ()=>{
  await Team.create({name: '火箭'});
  await Player.bulkCreate([{name: '哈登', teamId:1},{name: '保罗', teamId:1}]);

  // 1端关联查询
  const players = await Player.findAll({include:[Team]});
  console.log(JSON.stringify(players,null,2));

  // N端关联查询
  const team = await Team.findOne({where:{name:'火箭'},include:[Player]});
  console.log(JSON.stringify(team,null,2));
});

// 多对多关系
const Fruit = sequelize.define("fruit", { name: Sequelize.STRING });
const Category = sequelize.define("category", { name: Sequelize.STRING });
Fruit.FruitCategory = Fruit.belongsToMany(Category, {
  through: "FruitCategory"
});

// 插入测试数据
sequelize.sync({ force: true }).then(async () => {
  await Fruit.create(
    {
      name: "香蕉",
      categories: [{ id: 1, name: "热带" }, { id: 2, name: "温带" }]
    },
    {
      include: [Fruit.FruitCategory]
    }
  );
});

// 多对多联合查询
const fruit = await Fruit.findOne({
  where: { name: "香蕉" }, // 通过through指定条件、字段等
```

```
        include: [{ model: Category, through: { attributes: ['id', 'name']  
    } }]  
    });  
  })
```

