

## 13.MySQL高级开发下(数据库开发流程)✓

### 1.数据库Schema设计、开发流程

#### 1.0 为什么要进行数据库设计?

农村修改房子需求能住就行，功能要求不高，所以不需要设计。==小型数据库业务架构



修建茅屋需要设计吗？

城市修建大厦功能需求会更多，推到重新搭建代价太大，所以需要设计。==大型数据库业务架构



## 1.1 基本流程介绍

第一阶段：可行性，需求分析阶段。（产品经理，软件架构师）

第二阶段：概要设计阶段：E-R图、模型图（物理模型图）、需求确认（DBA）（开发环境）

第三阶段：详细设计阶段：范式、规范

第四阶段：代码编写阶段

第五阶段：功能测试阶段：功能完整性、性能、安全（测试环境）

第六阶段：部署、交付：架构设计、实施、调试、上线（DBA）

第七阶段：维保、支持、培训（DBA）

### 1.1.1 第一阶段：需求分析阶段

分析业务和数据处理需求

#### 1. 收集信息

与该系统有关人员进行沟通、交流、座谈，充分了解业务和数据处理需求。了解数据库要完成的任务。

例如：一个酒店、银行、医院、运营商、政企等单位的业务系统的基本功能。

酒店管理系统基本功能例子：

客户信息：客户基本信息、入住客房信息等。

客房信息：客房号、床位数、床位类型、价格等。

客房管理：客房号、床位类型、客房状态等。

财务管理

#### 2. 确认实体和属性

实体 (ENTITY)： 数据库要管理的实体或对象（表名）。

实体属性(ATTRIBUTE):标识每个实体的属性（列）

3.确认实体间关系

关系 (RELATIONSHIP)

## 1.1.2 第二阶段：概要设计阶段（DBA开始参与）

### 1. ER图与模型图的介绍

设计ER图，即是设计数据库的“图纸”，模型图即是将图纸进行物理化设计。方便与用户确认需求。



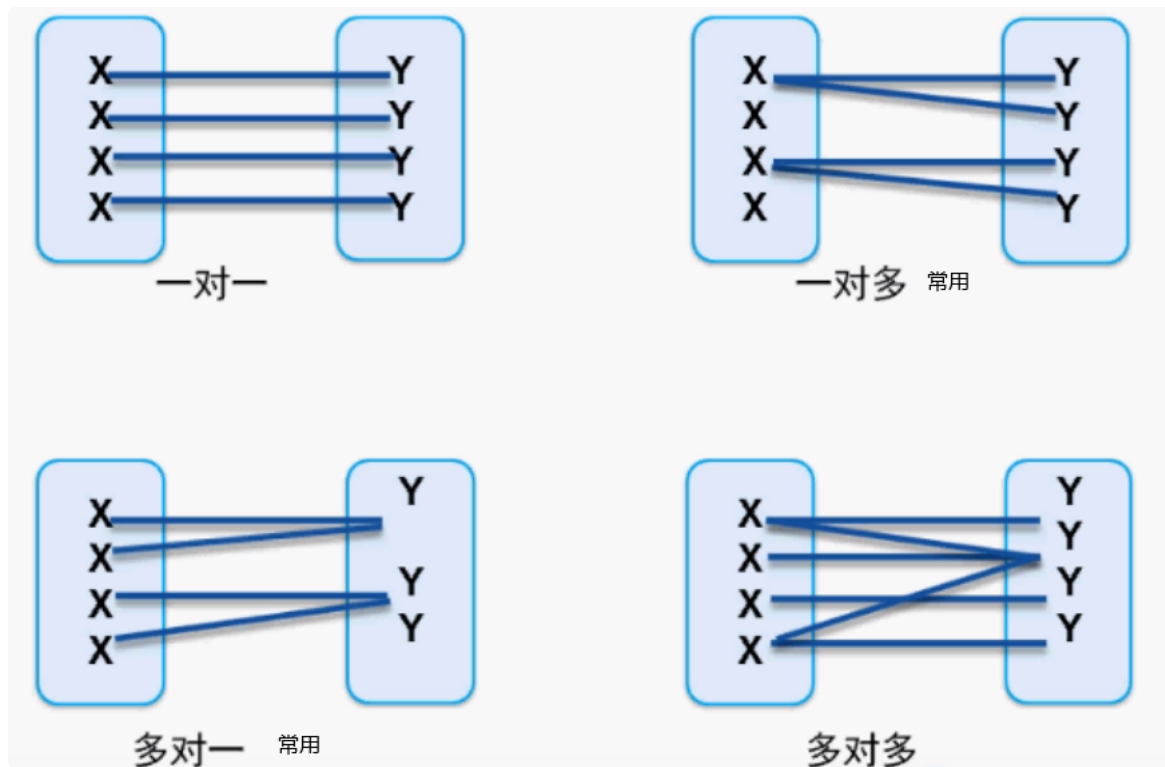
### 2. ER图常用绘画图形介绍

矩形----> 实体（表）

椭圆---->实体属性（列）

菱形---->表与表之间的关系

### 3.关系型数据库中表与表之间的映射关系



#### 4.根据需求信息绘制ER图

我们举个简单的例子，从第一阶段需求分析得到信息，设计学生与班级的ER图。

实体(表):        实体属性(列)

学生 (student) : sno sname sage sgender

班级 (class) : cno cname

关系 (表与表之间的关系) : 一对多

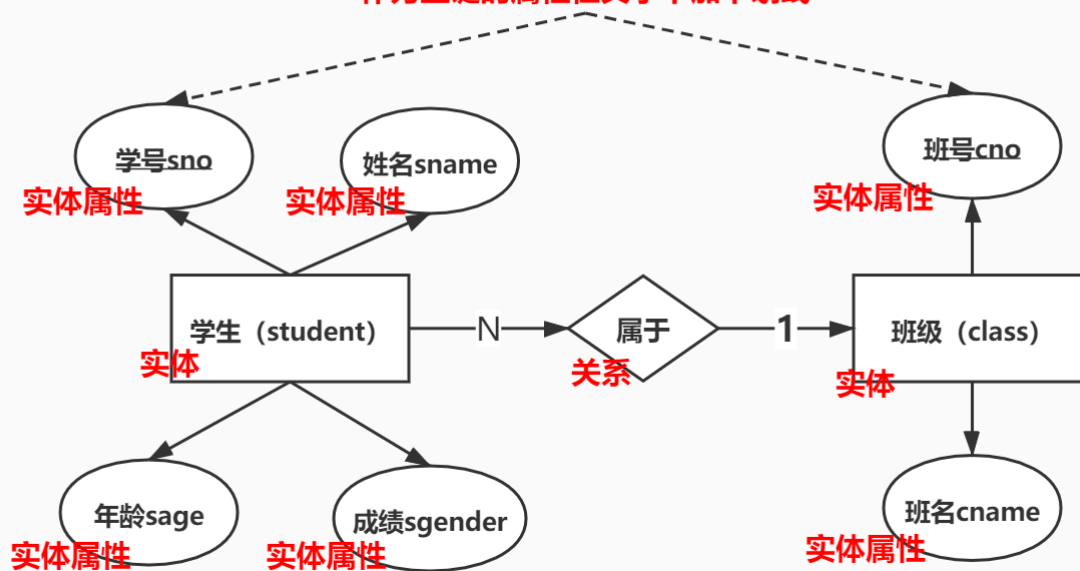
班级 (class) 1--->N 学生 (student)

我们可以使用processon画图工具里添加更多图形中的E-R图模板。把上面的需求信息用ER图展示出来。



## ER图-学生与班级

作为主键的属性在文字下加下划线



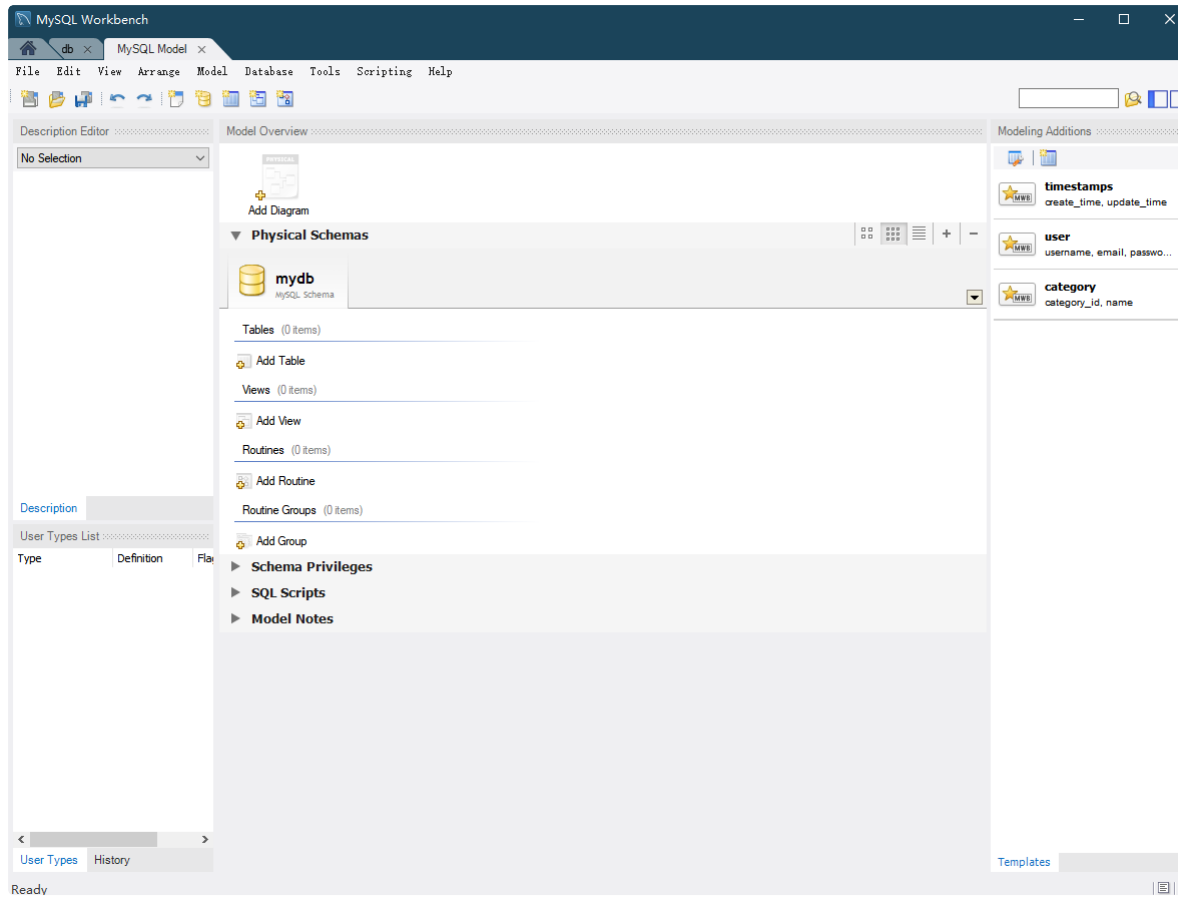
### 5.将E-R图转换成物理模型图(建模)

我们建模可以使用的工具

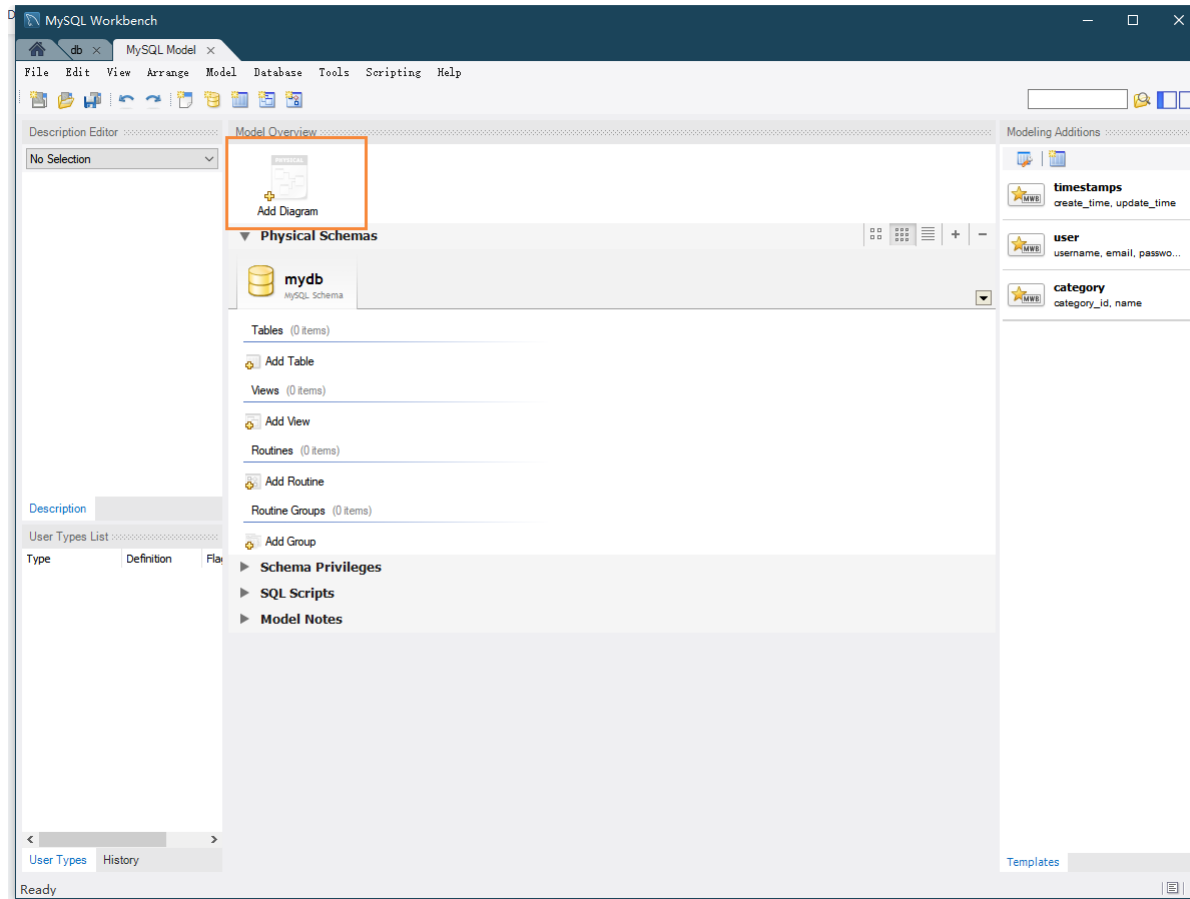
PD、Navicat、sqlyog、workbench

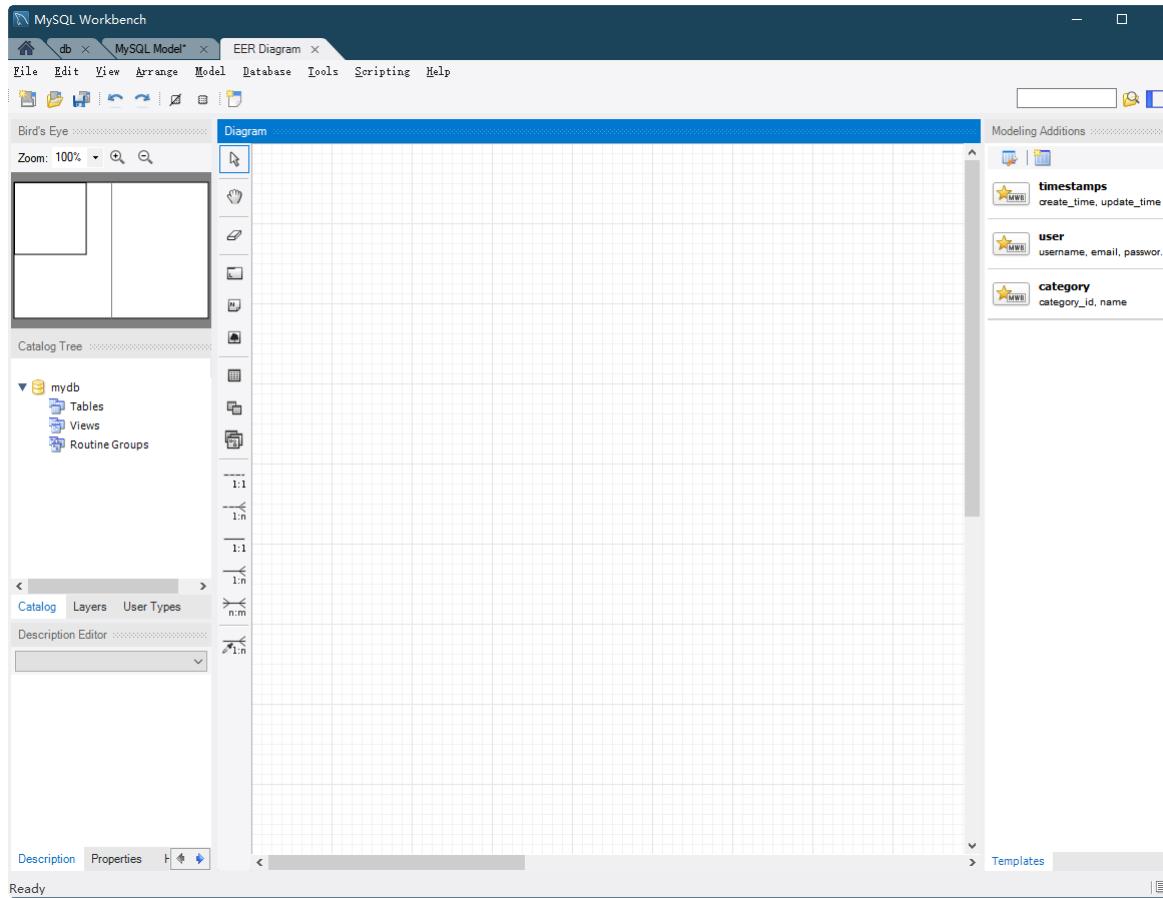
接下来我们使用workbench操作

#### 5.1.建立模型



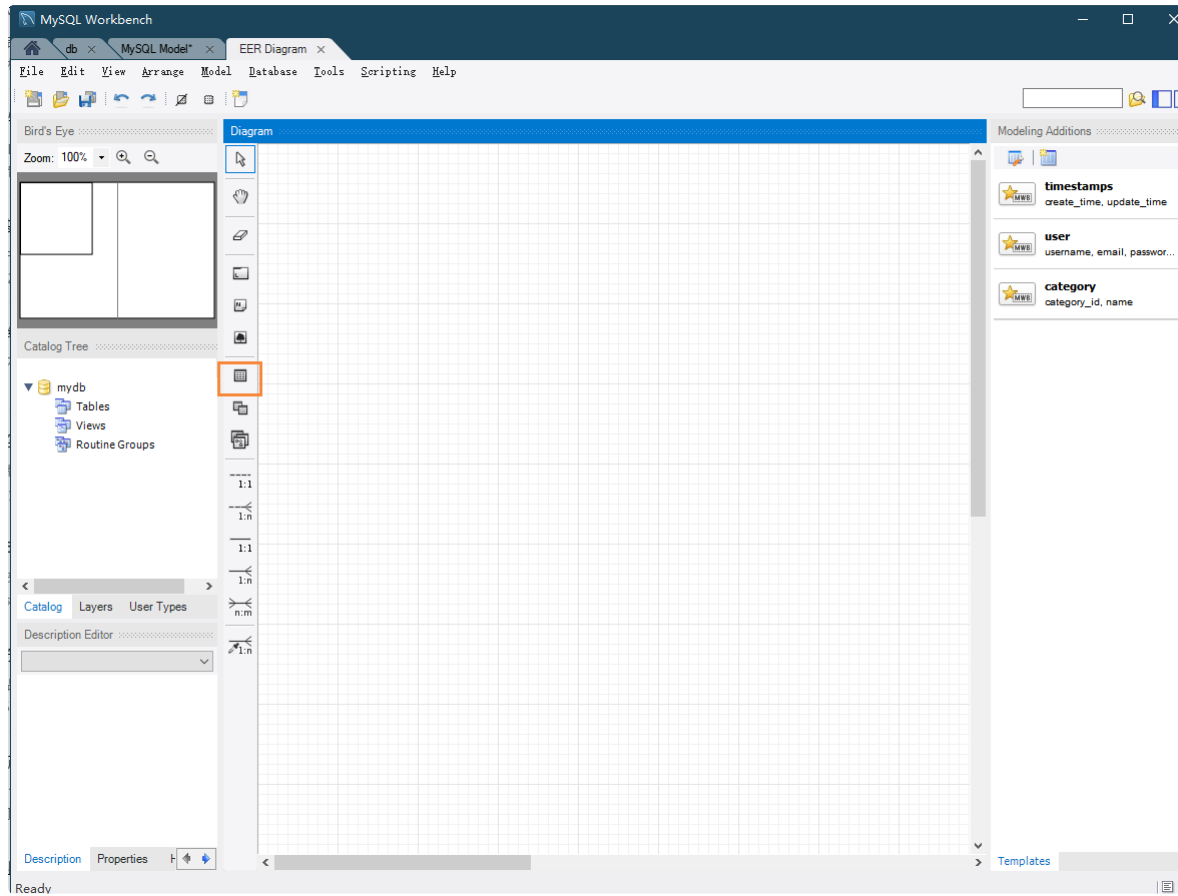
## 5.2 打开建模窗口

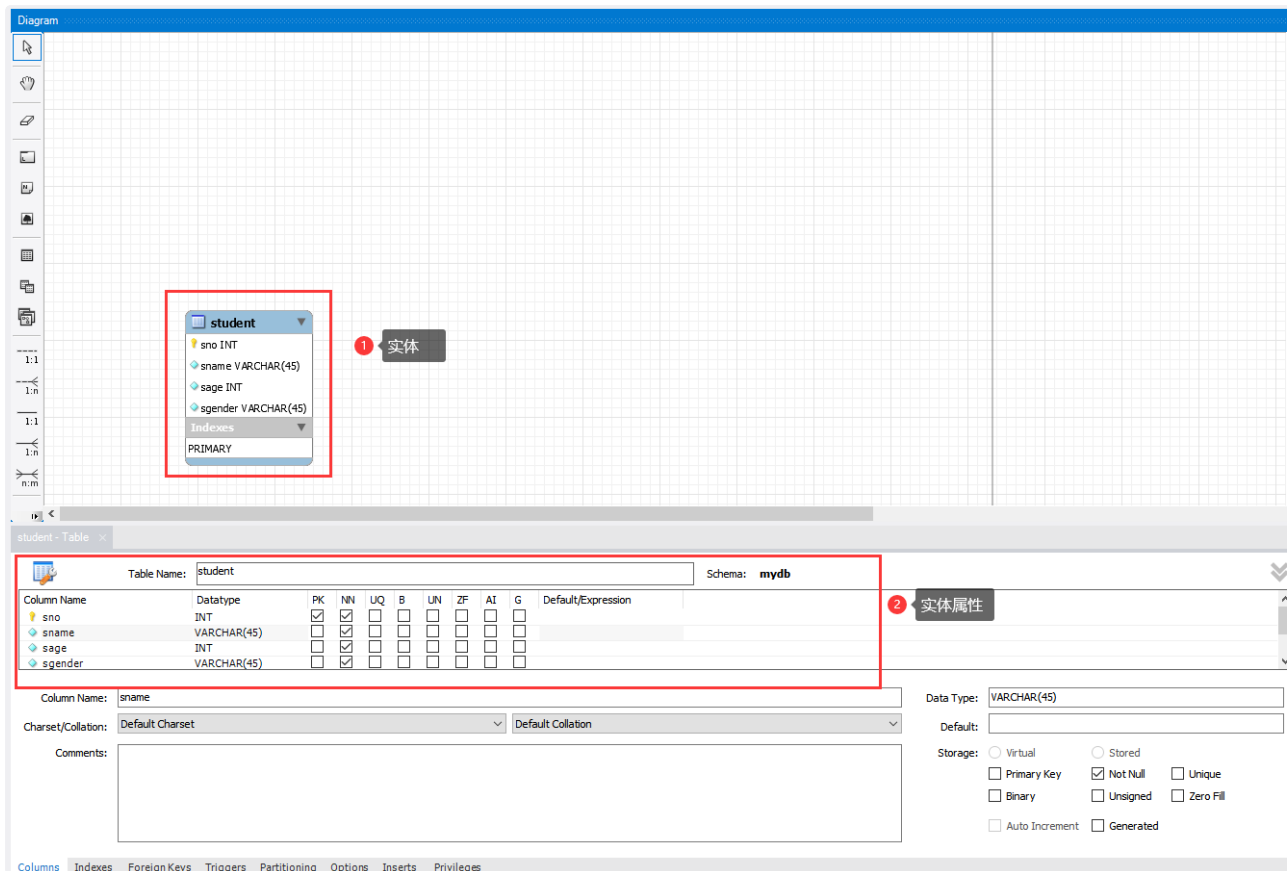




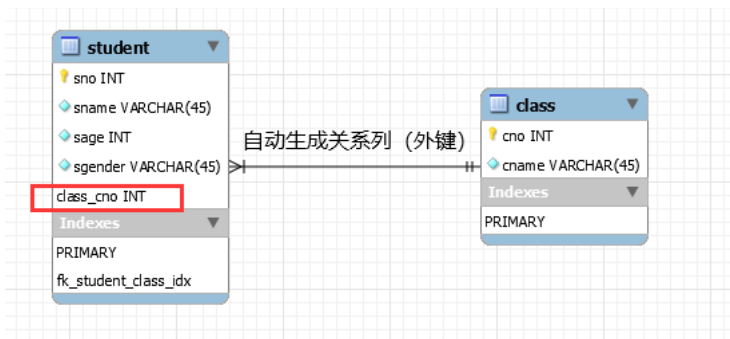
### 5.3 创建实体和实体属性



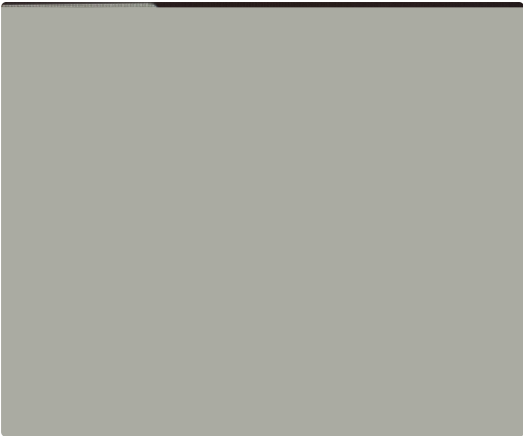




#### 5.4 建立表与表之间的关系（自动生成关系列）



#### 6.将建模的模型生成sql语句导入到数据库中



Synchronize Model with Database

Connection Options

Sync Options

Connect to DBMS

Select Schemas

Retrieve Objects

Select Changes to Apply

Review DB Changes

Synchronize Progress

Set Parameters for Connecting to a DBMS

Stored Connection:dbSelect from saved connection settings

Connection Method:Standard (TCP/IP)Method to use to connect to the RDBMS

ParametersSSLAdvanced

Hostname:10.0.0.51Port:3306Name or IP address of the server host - and TCP/IP port.

Username:oldguoName of the user to connect with.

Password:Store in Vault ...ClearThe user's password. Will be requested later if it's not set.

Default Schema:The schema to use as default schema. Leave blank to select it later.

Back

Next

Cancel

<https://www.yuque.com/kennethery/qzv4ul/ytfv4v>

11/25

Synchronize Model with Database

Connection Options

Sync Options

Connect to DBMS

Select Schemas

Retrieve Objects

Select Changes to Apply

Review DB Changes

Synchronize Progress

Set Options for Synchronization Script

Compare Options

☐ Skip synchronization of Triggers

☐ Skip synchronization of Stored Procedures and Functions

☐ Skip checking of Routine Definer

Generation Options

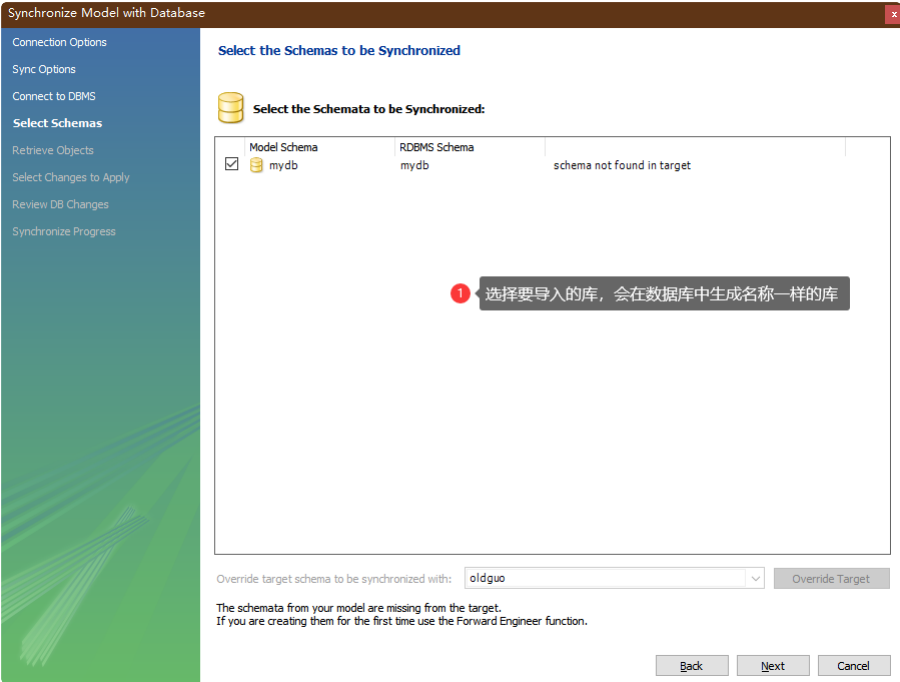
☐ Omit Schema Qualifier in Object Names

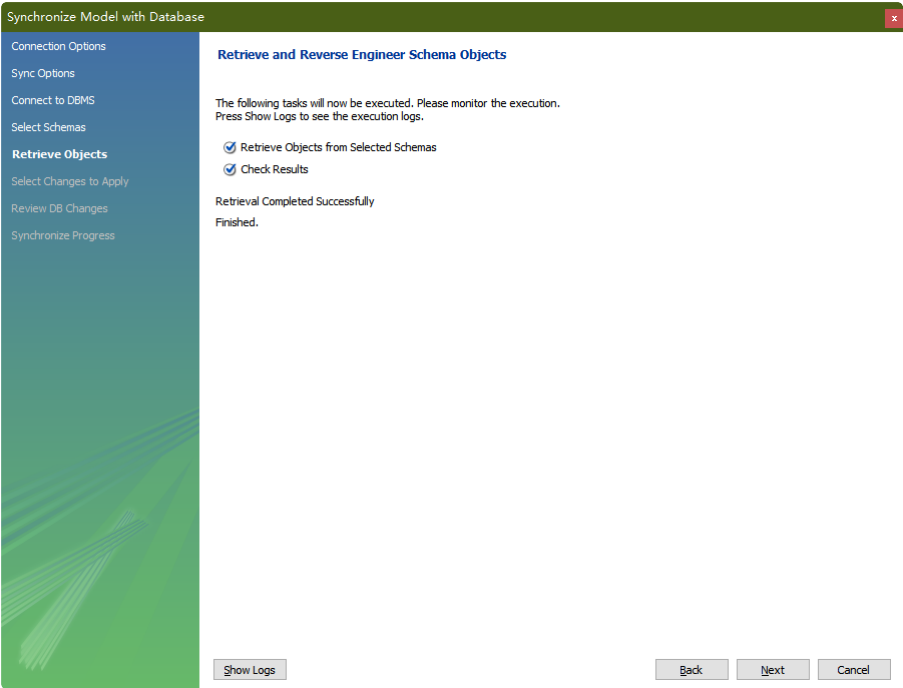
☐ Include SQL Scripts Attached to Model

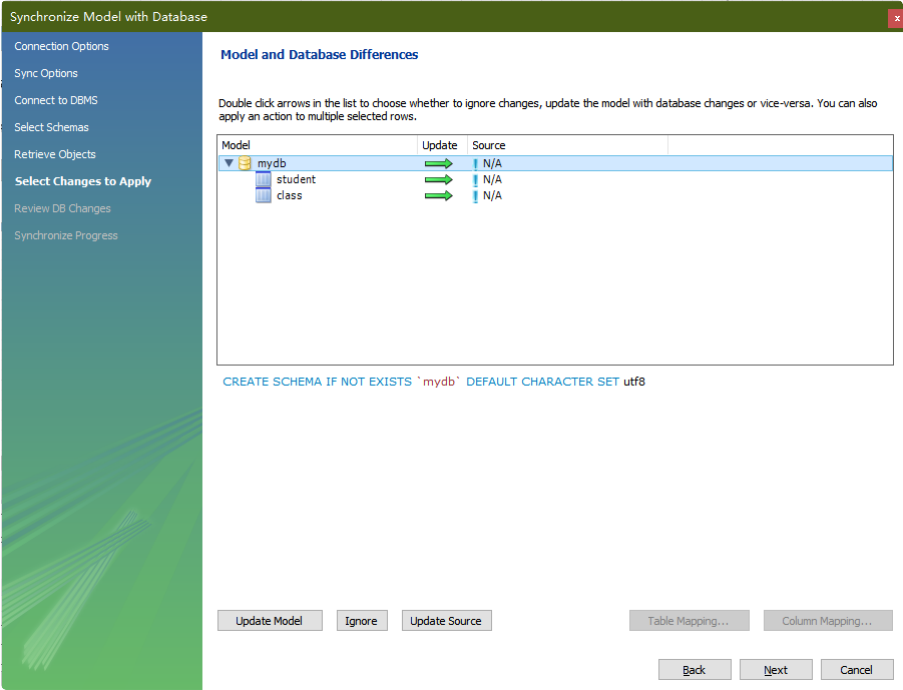
Back

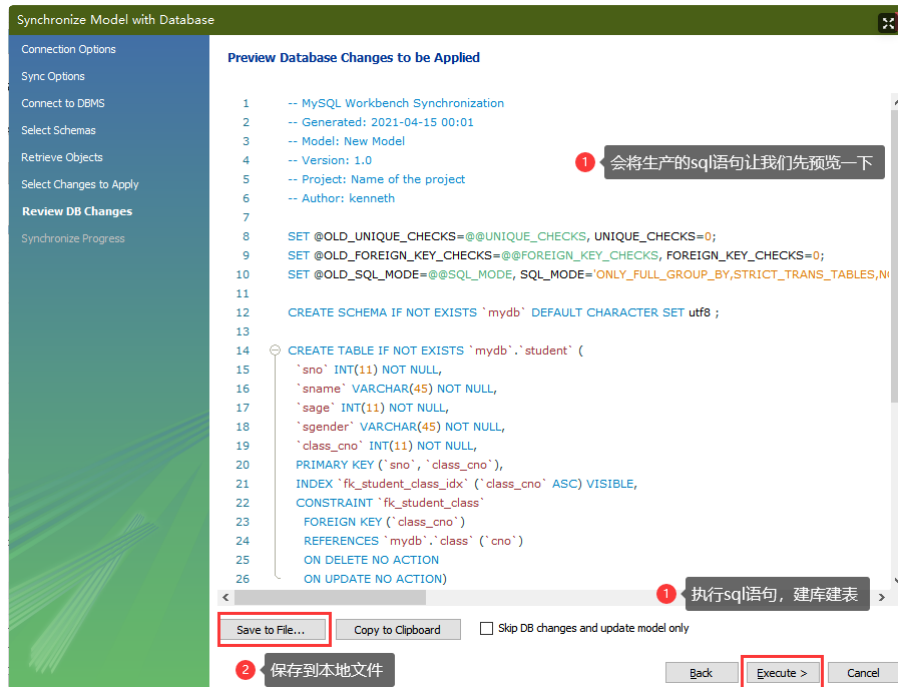
Next

Cancel

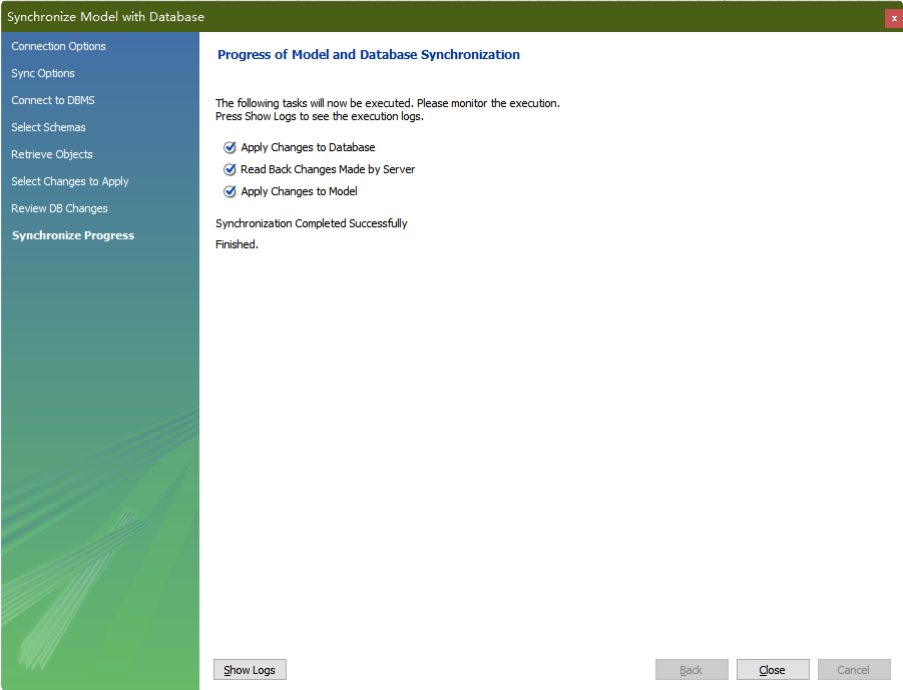




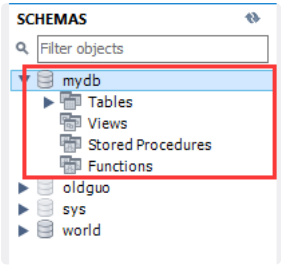








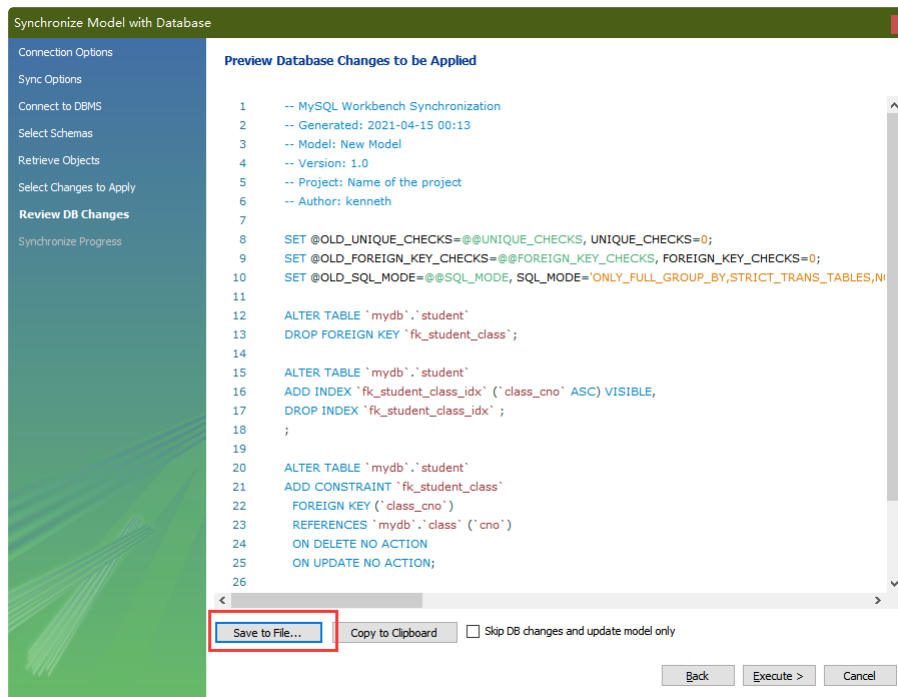
最后在数据库中生产我们建模设计的表



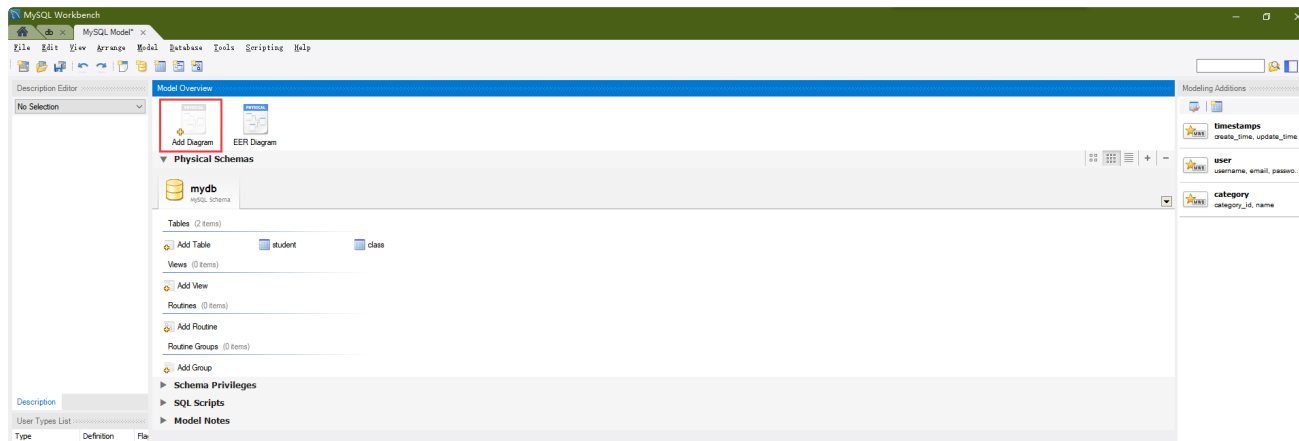
7.建模反操作

可以将sql语句转化成建模图形，前提是表与表之间要有外键关系。

1.我们这里把上面预览sql保存到本地，进行反操作

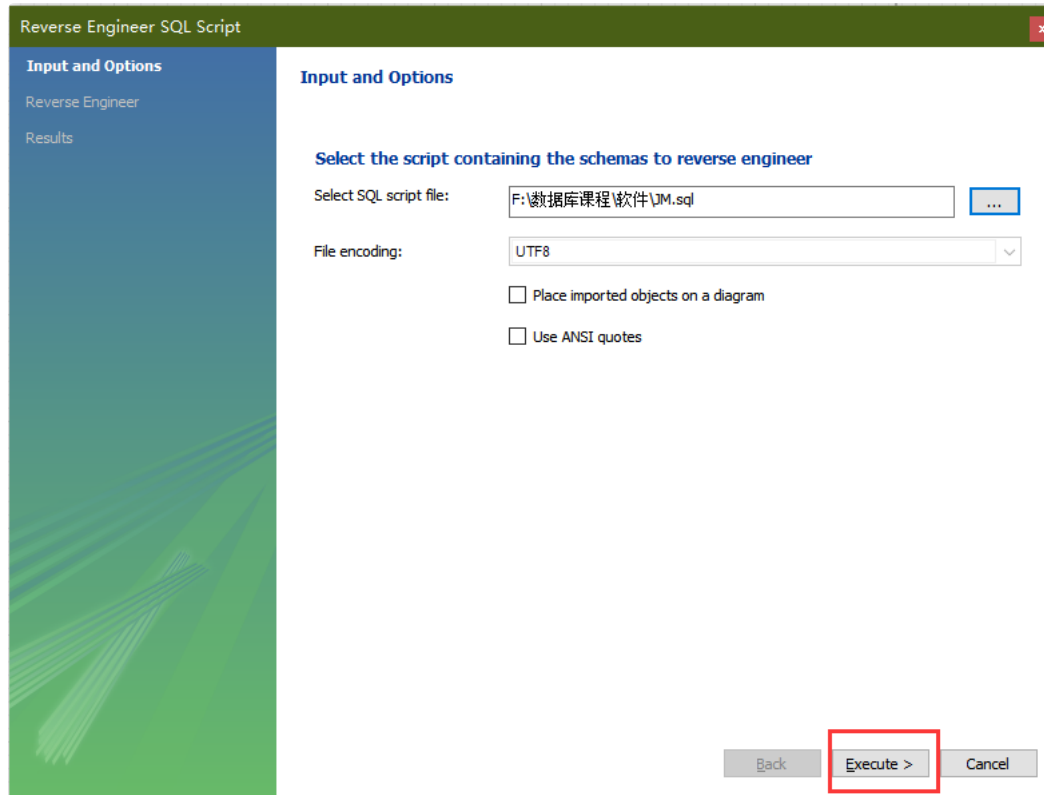


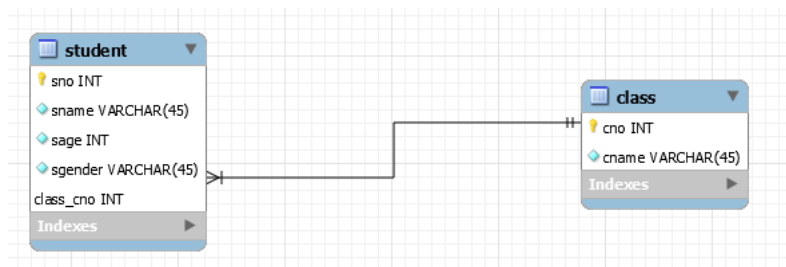
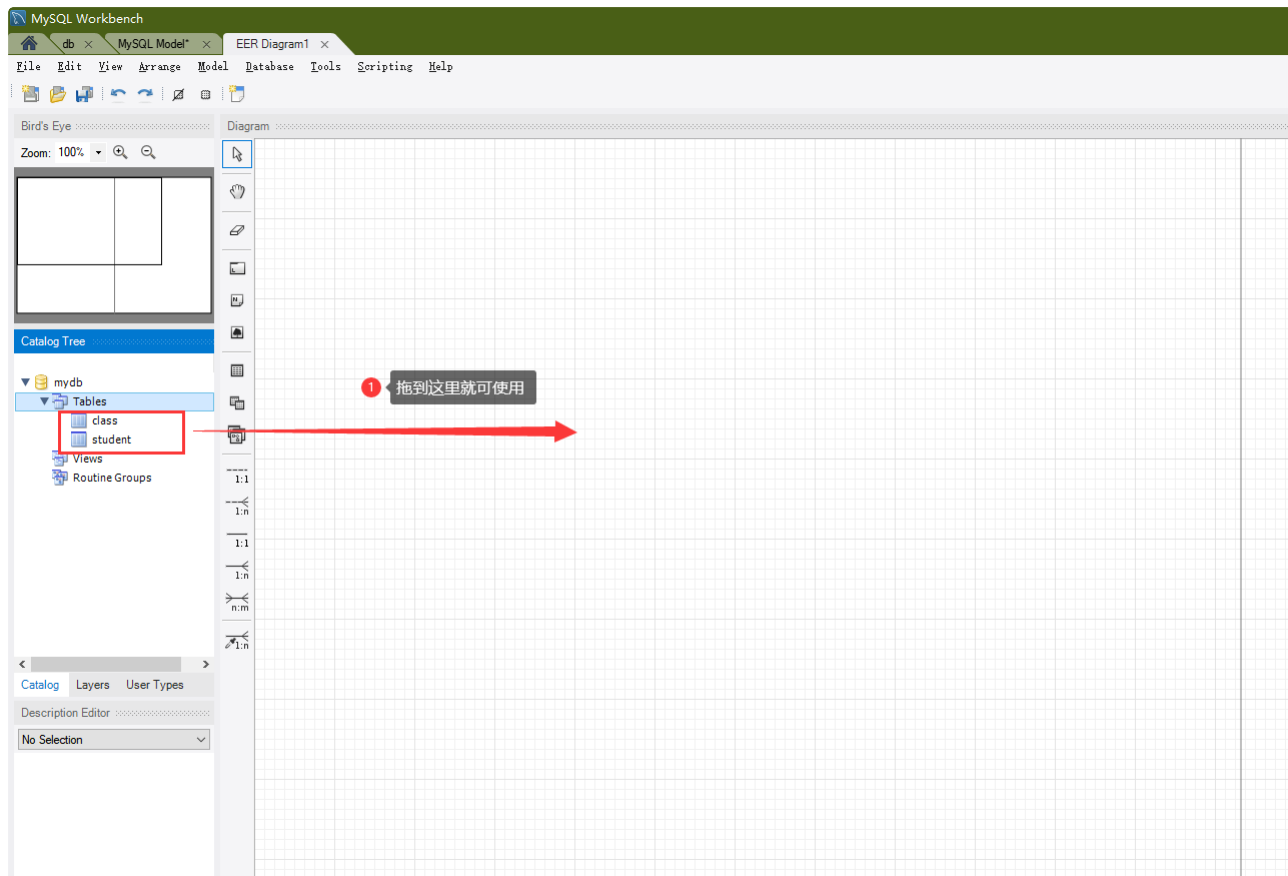
## 2.打开一个模板窗口



## 3.将sql语句导入进来，拖到空间中就可以用图的方式显示







### 1.1.3 第三阶段：详细设计阶段

#### 1.数据规范化的原因

保证数据的一致性和完整性。

不规范的设计会导致：

- 信息重复
- 数据量大
- 更新异常
- 插入异常
- 删除异常
- 查询异常

客人编号	姓名	地址	.....	客房号	客房描述	客房类型	客房状态	床位数	价格	入住人数
C1001	张三	Addr1	.....	1001	A栋1层	单人间	入住	1	128.00	1
C1002	李四	Addr2	.....	2002	B栋2层	标准间	入住	2	168.00	0
C1003	王五	Addr3	.....	2002	B栋2层	标准间	入住	2	168.00	2
C1004	赵六	Addr4	.....	2003	B栋2层	标准间	入住	2	158.00	1
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
C8006	A1	Addrm	.....	8006	C栋3层	总统套房	入住	3	1080.00	1
C8008	A2	Addrm	.....	8008	C栋3层	总统套房	空闲	3	1080.00	0

2.数据规范化的实现（关系型数据库的特点）

1NF:第一范式

原则：确保每个对象属性（列）的原子性 ,每列都是不可再分的最小原子单元。

原表

第一范式后的表

CustID	Address
1	中国北京市
2	美国纽约市
3	英国利物浦
4	日本东京市
.....	.....

CustID	Country	City
1	中国	北京
3	英国	利物浦
4	日本	东京
2	美国	纽约
.....	.....	.....

2NF:第二范式

每个表都只表示一种实体。只能描述一种实体的属性。

原表表示两种实体，客房信息+客户信息

第二范式后分开表示实体

## 示例 Guest Registration

字段	例子
客人编号	C1002
姓名	李四
地址	Addr1
客房号	2002
客房状态	入住
客房类型	标准间
床位数	2
入住人数	2
价 格	\$168.00
.....	.....

Guest



Room

字段	例子
客人编号	C1002
姓名	李四
地址	Addr1
客房号	2002

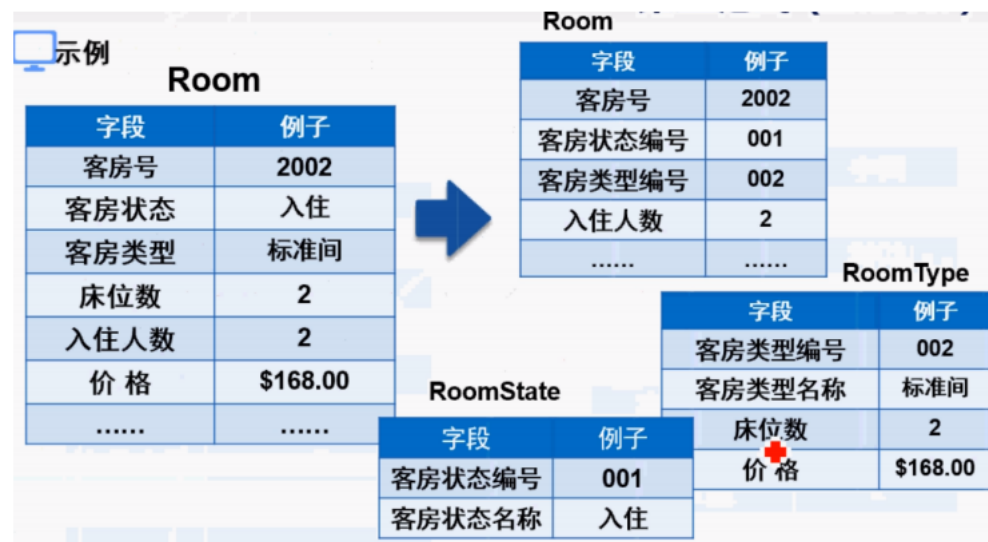
字段	例子
客房号	2002
客房状态	入住
客房类型	标准间
床位数	2
入住人数	2

### 3NF:第三范式 (解耦: 减少耦合性)

在满足以上其他范式的基础上, 所有列都应直接依赖主键, 不能间接 (传递) 依赖。

每个实体都有直接关联的主键列

在查询时会使用到join on 多表连接 影响查询速度



满足前三大范式即可, 当然还有后续的其他范式

3.范式规范化案例

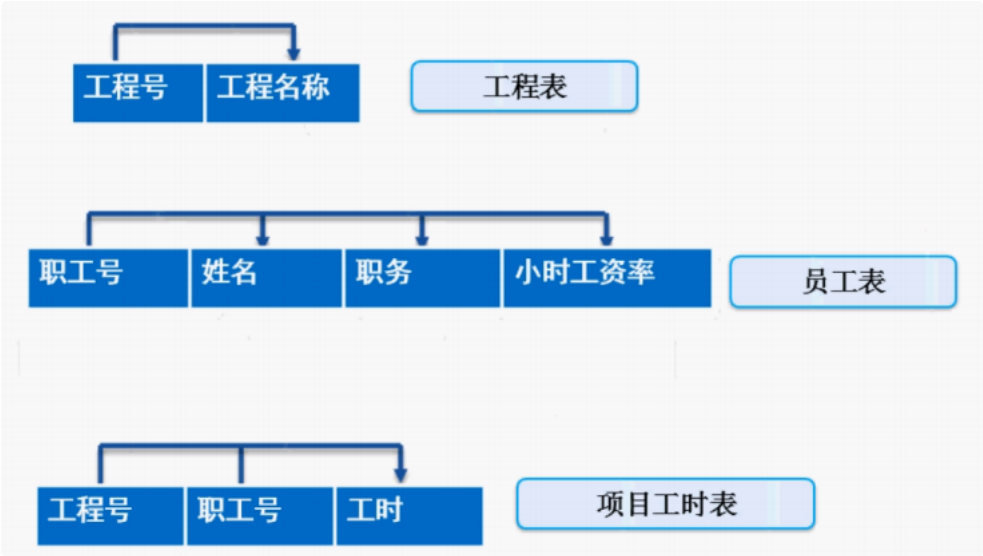
0.表中的小计和实发工资不需要设计，可以再调用这个表的数据信息时进行函数的运算

工程号	工程名称	职工号	姓名	职务	小时工资率	工时	实发工资
A1	花园大厦	1001	齐光明	工程师	65	13	845.00
		1002	李思岐	技术员	60	16	960.00
		1004	葛宇宏	sum函数，数据库中不需要设计			1140.00
			小计				2945.00
A2	立交桥	1001	齐光明	工程师	65	15	975.00
		1003	鞠明亮	工人	55	17	935.00
			小计				1910.00
A3	临江饭店	1002	李思岐	技术员	60	18	1080.00
		1004	葛宇洪	技术员	60	14	840.00
			小计				1920.00

1.上表已经满足第一范式，每个列信息不可再分割。

2.满足第二范式操作

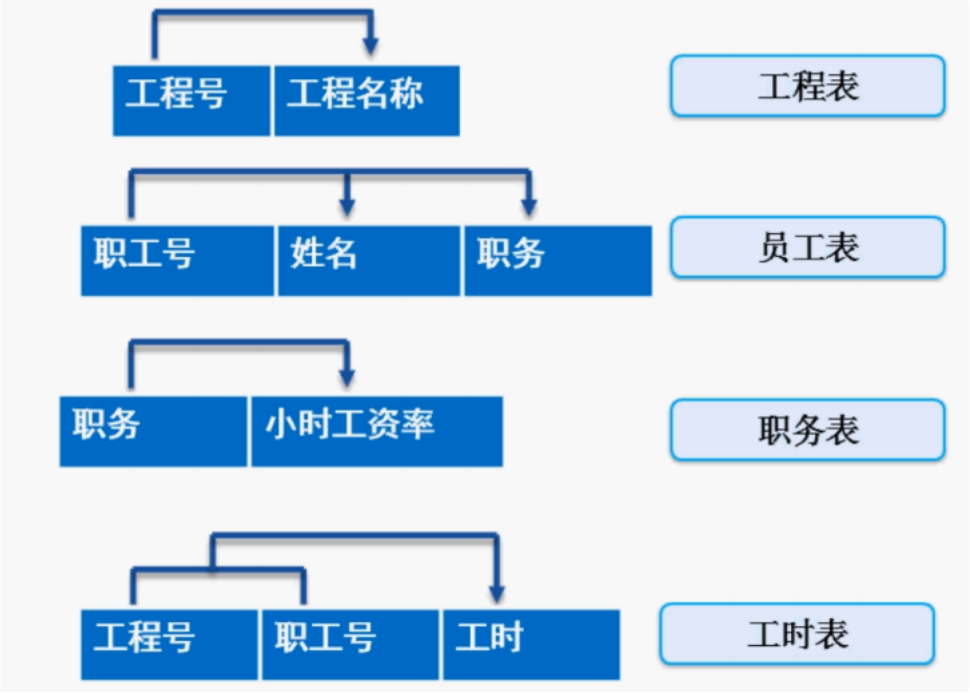
每个表表示一个实体，将原表的多实体拆分开来



3.满足第三范式操作



每个表中的所有列都有直接依赖的主键。



8d956f55e2f5.png&title=13.MySQL%E9%AB%98%E7%BA%A7%E5%BC%80%E5%8F%91%E4%B8%8B(%E6%95%B0%E6%8D%AE%E5%BA%93%E5%BC%80%E5%8F%91%E6%B5%81%E7%A8%8B)%E2%8