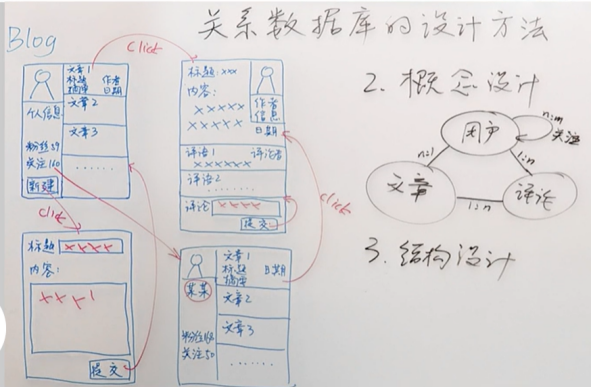
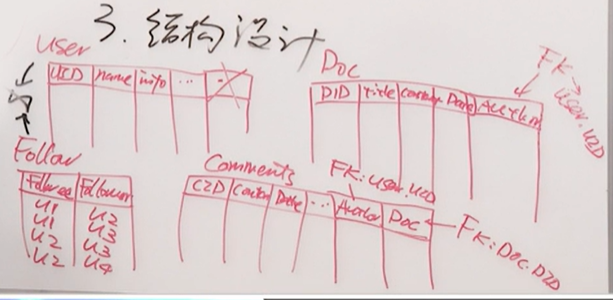
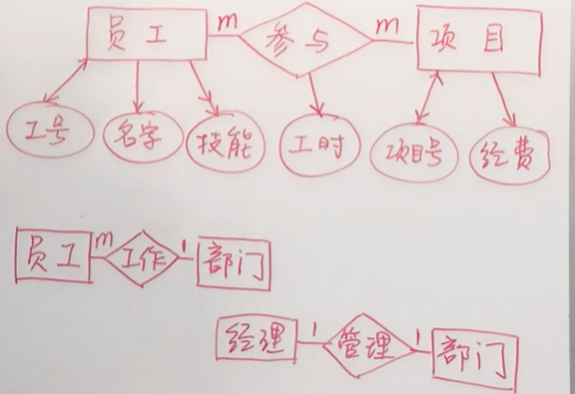
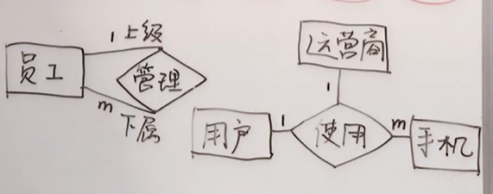
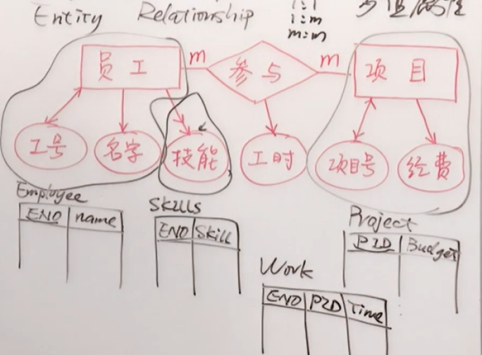
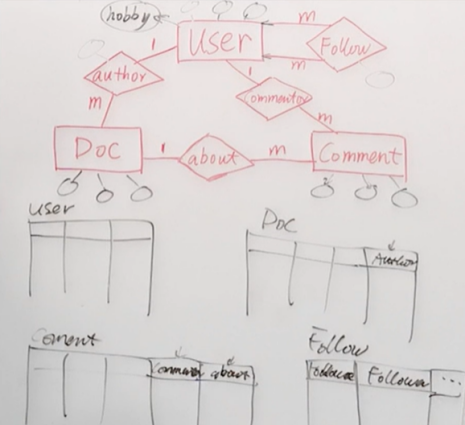
* 关系数据库设计
  1. 
* 博客实例的关系模式设计
  1. 
* ER图 / 实体联系图
  1.  
  2. 双向箭头：唯一属性，单值属性：单向箭头，多值属性：单向双箭头
  3. 一对多，一对一，多对多
* 从ER图到关系模式设计
  1. 
* ER图在博客实例中的应用
  1. 

**第*1*题：在ER图中有一个叫Person的实体类，它的属性分别为：ID（唯一属性）、Name（单值属性）、Hobby（多值属性）、Skills（多值属性）。对该实体类进行关系模式设计，得到的结果应该是：**

A：Person ( ID, Name, Hobby, Skills )

B：Person ( ID, Name ) 、PersonHS ( ID, Hobby, Skills )

C：Person ( ID, Name ) 、PersonH ( ID, Hobby ) 、PersonS ( ID, Skill ) √ 多值属性需要单独作为一张表

D：Person ( ID ) 、Person ( ID, Name ) 、PersonH ( ID, Hobby ) 、PersonS ( ID, Skill )

多值属性需要单独作为一张表的原因：

规范化

避免数据重复（如果将多值属性直接存储在包含其他属性的同一张表中，可能导致数据冗余。例如，如果某个实体有多个电话号码，将这些电话号码放在单独的表中，可以避免在主表中重复存储其他属性）

更好的查询性能（如果需要查找包含特定多值属性的实体，可以直接查询多值属性表，而不必搜索包含其他属性的主表）

支持关联（使得在查询中可以方便地使用 JOIN 操作，从而获取更复杂的数据）

可维护性

**第*2*题：一个ER图中有学生和课程两类实体。假设一门课每学期都可以开一个班，而学校规定一个学生可以重复多次选修同一门课，并获得多次不同的成绩。以下关于“学生选课”的ER图设计，说法最正确的是：**

A：将“选课”作为学生和课程之间的一类联系，该类联系对应的关系表的主码包括两个属性：学号和课程号。

B：需增加“开课班”这一个新的实体类，它与课程之间是多对一的联系。而将“选课”作为学生和开课班之间的一类联系，该类联系对应的关系表的主码包括两个属性：学号和开课班ID。

C：需增加“学期”这一个新的实体类。而将“选课”作为学生、课程和学期三者之间的一类联系，该类联系对应的关系表的主码包括三个属性：学号、课程号和学期编号

D：B和C均合理 √ 不能由两个实体唯一确定的联系不能作为联系，应作为实体

**第*3*题：在ER图表示的世界中，每个实体都应该有一个ID，这个ID可以是它的一个唯一属性，比如身份证号，也可以由它的多个单值属性组成，比如电影票上的场次和座位号。如果没有这样的ID，该类实体对应的关系表就缺少主键。但对于联系，是否也应该有这样的ID？如果有，这个ID是什么？**

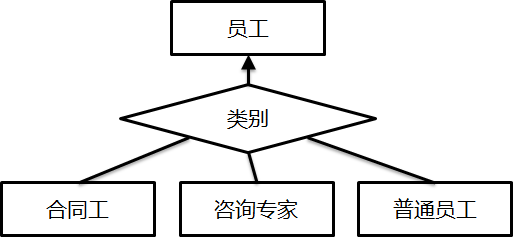
A：一个联系的ID应该由参与联系的所有实体ID共同组成 √ 联系是由实体来识别的

B：一个联系的ID应该由它的属性构成

C：一个联系的ID可以由它的属性以及参与联系的实体ID共同构成

D：一个联系不需要有ID

1. 下图所示的ER图刻画了四类实体，对“员工”而言，另三类实体（合同工、咨询专家和普通员工）都是它的子类。也就是说，员工拥有的属性，另三类实体都拥有；而子类拥有的属性父类未必拥有。请尝试设计这四类实体对应的关系表。



Staff(ID, Name, Position, …)

Contractor(StaffID, …)

Expert(StaffID, …)

Employee(StaffID, …)

**父类表：员工（Employee）**

CREATE TABLE Employee (

employee\_id INT PRIMARY KEY,

name VARCHAR(255),

hire\_date DATE,

-- 其他共有的属性

);

**子类表1：合同工（Contractor）**

CREATE TABLE Contractor (

contractor\_id INT PRIMARY KEY,

employee\_id INT,

contract\_duration INT,

hourly\_rate DECIMAL(10, 2),

FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id),

-- 其他特有的属性

);

**子类表2：咨询专家（Consultant）**

CREATE TABLE Consultant (

consultant\_id INT PRIMARY KEY,

employee\_id INT,

expertise\_area VARCHAR(255),

hourly\_rate DECIMAL(10, 2),

FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id),

-- 其他特有的属性

);

**子类表3：普通员工（RegularEmployee）**

CREATE TABLE RegularEmployee (

regular\_employee\_id INT PRIMARY KEY,

employee\_id INT,

department VARCHAR(255),

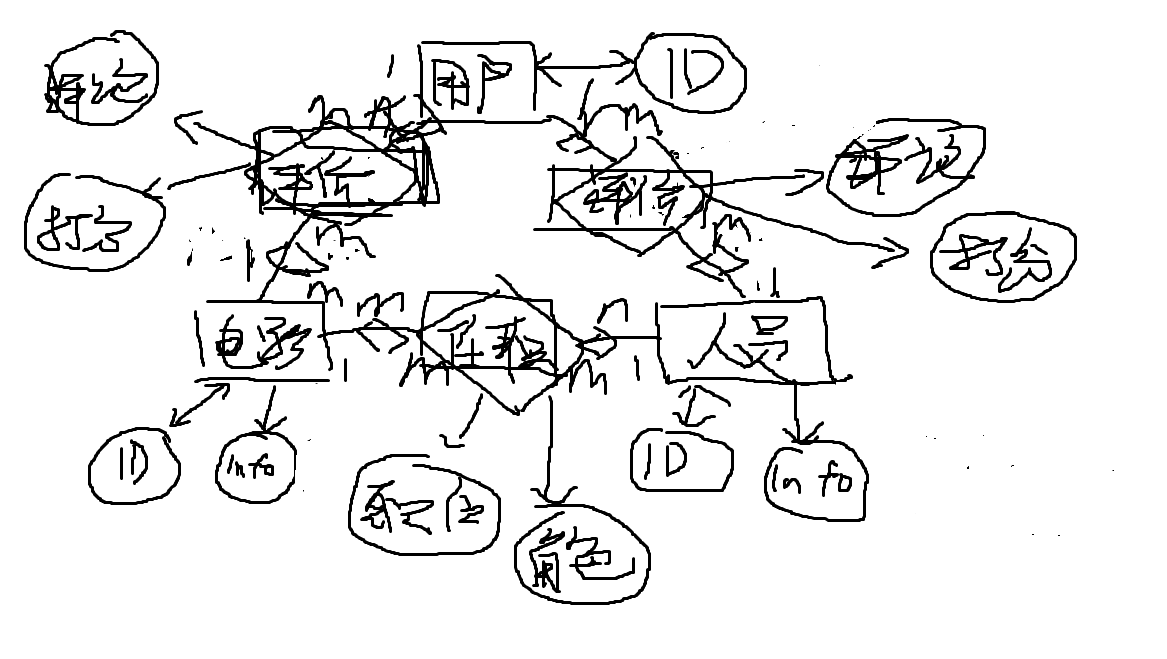
position VARCHAR(255),

FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id),

-- 其他特有的属性

);

1. 请针对以下需求设计ER图，并构建相应的关系模式：  
   一个关于电影、制作人员和演员的网站（类似一个简易的IMDB网站）。用户可以浏览每一部电影的简介和相关信息（出品年份、电影类型、时长、评级），以及其导演、编剧和演员的列表。用户还可以浏览每一位导演、编剧或演员的信息（姓名、性别、年龄、简介），以及他们参与过哪些电影作品。导演、编剧或演员只是职位，一个人可以身兼数职。  
   每一位演员在其参演的电影中都扮演一定的角色。用户在浏览电影时，除了能看到演员信息，还能看到每位演员扮演了什么角色。用户在浏览演员时，除了能看到他（或她）参演的电影，还能看到他（或她）在每一部电影中扮演的角色。（注意：一位演员可以在一部电影中扮演多个角色。同一个角色也可能由多名演员扮演，比如，年少时由一位演员扮演，年老时由另一位演员扮演。）  
   用户登录后还可以针对每一部电影、每一位导演、编剧或演员进行评价和打分，供别人参考。



Monie (ID, Intro, Info)

Staff (ID, Info)

User (ID)

Work (ID, MID, SID, position, character)

EvaluationM (ID, UID, MID, Comment, Score)

EcaluationS (ID, UID, SID, Comment, Score)