

选择题

1. 设关系 R 和 S 的元组个数分别为 100 和 300, 关系 T 是 R 与 S 的笛卡尔积, 则 T 的元组个数为 ()。
A、400
B、300
C、30000
D、90000
2. 在数据库的表定义中, 限制成绩属性列的取值在 0 到 100 的范围内, 属于数据的 () 约束。
A、实体完整性
B、参照完整性
C、用户定义的完整性
D、用户操作
3. 以下哪一个不是数据模型的组成要素 ()。
A、数据
B、数据结构
C、数据操作
D、完整性约束
4. 学生与学号之间的联系是 ()。
A.一对一
B.一对多
C.多对多
D.多对一
5. 假设有如下关系模式: 学生选课记录表(学生编号, 课程编号, 学生姓名, 学生年龄, 课程名称, 课程学分, 选修成绩) 请问该关系模式满足 () 范式。
A、仅满足第一范式
B、满足第二范式
C、满足第三范式
D、都不满足
6. 在子模式是用户与数据库的接口之一, 它对应于数据库的 ()。
A. 外模式
B. 概念模式
C. 内模式
D. 存储模式
7. 如果 E-R 图中实体 A 和实体 B 的联系是 m:n 关系, 以下说法正确的是 ()。
A、将该联系单独转换成一个关系模式

- B、将其中一个实体的码合并到另一个实体对应的关系模式中，称为外码
- C、直接将两个实体的关系模式合并
- D、以上都不正确

8. 关于索引的说法，不正确的是（ ）。

- A、索引加快数据的检索速度
- B、索引不需要占据物理空间
- C、创建唯一性索引，保证数据库表中每一行数据的唯一性
- D、当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，降低了数据的维护速度

9. 若事务 T 对数据对象 A 加上 S 锁，则（ ）。

- A、事务 T 可以读 A 和修改 A，其它事务只能再对 A 加 S 锁，而不能加 X 锁
- B、事务 T 可以读 A 但不能修改 A，其它事务只能再对 A 加 S 锁，而不能加 X 锁
- C、事务 T 可以读 A 但不能修改 A，其它事务能对 A 加 S 锁和 X 锁
- D、事务 T 可以读 A 和修改 A，其它事务能对 A 加 S 锁和 X 锁

10. 若系统在运行过程中，由于某种硬件故障，使存储在外存上的数据部分损失或全部损失，这种情况称为（ ）。

- A、事务故障
- B、系统故障
- C、介质故障
- D、运行故障

11. 以下说法不正确的是（ ）。

- A、具有相同类型的值的集合，称为域，教师集合可以表示为 $(T) = \{t1, t2\}$ ，学生的集合可以表示为： $(S) = \{s1, s2, s3\}$
- B、一个关系的多个候选码中选定一个作为主码，比如：部门编号，唯一的确定一个部门
- C、关系的描述称作关系模式
- D、学生（学号，姓名，年龄，性别，系，年级）描述的是一个关系

12. 要从学生关系中查询学生的姓名和性别，需要进行的关系运算是（ ）。

- A、选择
- B、投影
- C、连接
- D、求交

13. 当系统运行过程中发生故障，针对不同的故障可选择不同的恢复策略。下面不正确的恢复策略是（ ）。

- A. 事务故障的恢复
- B. 系统故障的恢复
- C. 介质故障的恢复

D . 生态故障的恢复

14. 在数据库设计中, 用 E-R 图来描述信息结构但不涉及信息在计算机中的表示, 它是数据库设计的 () 阶段。

A、需求分析阶段

B、概念设计阶段

C、逻辑设计阶段

D、物理设计阶段

15. 根据关系数据库规范化理论, 关系数据库中的关系要满足第一范式。在客户(客户 ID, 客户名, 公司名, 联系方式)关系中, 哪个属性可能使它不满足第一范式? ()

A、客户 ID

B、客户名

C、公司名

D、联系方式

填空题

数据库语言

事务

数据库系统

完整性

SQL 查询

存储

锁

关系模式和范式

数据库恢复

简答题

1. 什么是封锁？

封锁就是事务 T 在对某个数据对象例如表、记录等操作之前，先向系统发出请求，对其加锁。加锁后事务 T 就对该数据对象有了一定的控制，在事务 T 释放它的锁之前，其他的事务不能更新此数据对象。封锁是实现并发控制的一个非常重要的技术。

2. 数据库恢复的基本技术有哪些？

数据转储和登记日志文件是数据库恢复的基本技术。

当系统运行过程中发生故障，利用数据库后备副本和日志文件就可以将数据库恢复到故障前的某个一致性状态。

3. 什么是数据库转储？

数据转储是数据库恢复中采用的基本技术。所谓转储即 DBA 定期地将数据库复制到磁带或另一个磁盘上保存起来的过程。当数据库遭到破坏后可以将后备副本重新装入，将数据库恢复到转储时的状态。

4. 数据库的完整性概念与数据库的安全性概念有什么区别？

(1) 前者是为了防止数据库中存在不符合语义的数据，防止错误信息的输入和输出，即所谓垃圾进垃圾出所造成的无效操作和错误结果。

(2) 后者是保护数据库防止恶意的破坏和非法的存取。也就是说，安全性措施的防范对象是非法用户和非法操作，完整性措施的防范对象是不合语义的数据。

5. 为什么事务非正常结束时会影响数据库数据的正确性？

事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。如果数据库系统运行中发生故障，有些事务尚未完成就被迫中断，这些未完成事务对数据库所做的修改有一部分已写入物理数据库，这时数据库就处于一种不正确的状态，或者说的不一致的状态。

6. 设 R 和 S 两关系如下：求 $R \bowtie S$ 并写出求解过程。

R			S	
A	B	C	B	C
a2	b1	c2	b1	c2
a3	b3	c7	b2	c1
a2	b2	C1		
a3	b4	c6		
a1	b2	c3		
a1	b1	c2		
a1	b2	c1		

R ∞ S

A	B	C
a1	b1	c2
a1	b2	c1
a2	b2	c3
a1	b2	c3

7. 如何用封锁机制保证数据的一致性？

DBMS 在对数据进行读、写操作之前首先对该数据执行封锁操作，例如事务 T1 在对 A 进行修改之前先对 A 执行 lock(A)，即对 A 加 x 锁。这样，当其他请求对 A 加 x 锁时就被拒绝，只能等待 T1 释放 A 上的锁后才能获得对 A 的 x 锁，这时它读到的 A 是 T1 更新后的值，再按此新的 A 值进行运算。这样就不会丢失 T1 的更新。

DBMS 按照一定的封锁协议，对并发操作进行控制，使得多个并发操作有序地执行，就可以避免丢失修改、不可重复读和读“脏”数据等数据不一致性。

语言描述题

设有商店和顾客两个实体，假设一个商店有多个顾客购物，一个顾客可以到多个商店购物，顾客每次去商店购物有一个消费金额和日期，而且规定每个顾客在每个商店里每天最多消费一次。有如下关系表用于描述顾客去商店消费的情况，关系中有下划线的表示主码。

Customer (Cno, Cname, Csex, Cage)

顾客表：表示顾客编号、姓名、年龄、性别

Store (Sno, Sname, Saddress, Sphone)

商店表：表示商店编号、商店名、地址、电话

Spend (Sno, Cno, Samount, Sdate)

消费表：表示商店编号，顾客编号，消费金额，日期

请用关系代数及 SQL 语句完成如下题目：

1. 用关系代数实现去 1 号商店消费的顾客编号。

$\pi_{Cno}(\sigma_{Sno='1'}(Spend))$

2. 用关系代数实现女性的消费情况。

$\pi_{Cno}(\sigma_{Csex='女'}(Customer)) \bowtie Spend$

3. 用关系代数实现在 1 号商店消费并且消费金额在 800 以上(不含 800)的顾客编号和姓名。

$\pi_{Cno}(\sigma_{Sno='1' \wedge Samount > 800}(Spend)) \bowtie \pi_{Cno, Cname}(Customer)$

4. 用 SQL 语句实现在 1 号商店消费并且消费金额在 800 以上(不含 800)的顾客编号和姓名。

Select Customer.Cno,Cname

From Customer,Spend

Where Customer.Cno=Spend.Cno and Spend.Sno='1' and Spend.Samount>800

5. 用 SQL 语句实现对 Spend 表进行创建，注意主码、外码表示，其中 Cno 为 9 字符,Sno 为 4 字符, Samount 为长整数,Sdate 为日期。

CREATE TABLE Spend

(

Cno char (9),

Sno char (4),

Samount int,

Sdate date,

PRIMARY KEY (Cno, Sno),

FOREIGN KEY (Cno) REFERENCES Customer (Cno),

FOREIGN KEY (Sno) REFERENCES Store (Sno)

)

6. 用 SQL 语句查询顾客编号为“vip2019007”的顾客在 1 号商店的累计消费金额。

SELECT sum(Samount)

FROM Spend

WHERE Cno='vip2019007' and Sno = '1'

7. 用 SQL 语句查询在同一家商店购物超过 3 次（不含 3 次）的顾客编号。

```
SELECT Cno  
FROM Spend  
GROUP BY Cno  
HAVING COUNT(*)>3
```

8. 用 SQL 语句将商店编号为“7”的商店名称修改为“最美的你”。

```
UPDATE Store  
SET Sname ='最美的你'  
WHERE Sno ='7'
```

9. 创建一个女性顾客视图 F_S_VIEW，包括顾客编号、姓名、性别。

```
CREATE VIEW F_S_VIEW  
AS  
SELECT Cno,Cname,Cage  
FROM Customer  
WHERE Ssex='女'
```


设有运动员和比赛项目两个实体，一个运动员可以参加多个项目，一个项目由多名运动员参加，运动员参赛还包括比赛时间、比赛成绩等信息，有如下关系表用于描述运动员参加比赛的情况，关系中有下划线的表示主码。

Athlete (Ano, Aname, Asex, Aunit, Aage)

运动员：表示运动员编号、姓名、性别、单位、年龄

Game (Gno, Gname, Gresult)

比赛项目：表示比赛项目号、名称、历史最好成绩

Competition (Ano, Gno, Cgrade, Crank)

参赛表：表示运动员编号、比赛项目号、比赛成绩、比赛名次

请用关系代数及 SQL 语句完成如下题目：

1. 用关系代数实现参加了比赛项目号为 1 号的运动员的编号。

$$\pi_{\text{Ano}}(\sigma_{\text{Gno}='1'}(\text{Competition}))$$

2. 用关系代数实现惠州学院 (hzu) 的运动员参加比赛的情况。

$$\pi_{\text{Ano}}(\sigma_{\text{Aunit}='hzu'}(\text{Athlete})) \bowtie \text{Competition}$$

3. 用关系代数实现参加了比赛项目号为 1 号并且比赛名次为第一的运动员的编号和姓名。

$$\pi_{\text{Ano}}(\sigma_{\text{Gno}='1' \wedge \text{Crank}=1}(\text{Competition})) \bowtie \pi_{\text{Ano}, \text{Aname}}(\text{Athlete})$$

4. 用 SQL 语句实现参加了比赛项目号为 1 号并且比赛名次为第一的运动员的编号和姓名。

```
Select Athlete.Ano,Aname From Athlete,Competition
```

```
Where Athlete.Sno=Competition.Sno and Competition.Ano='1' and Competition.Crank=1
```

5. 用 SQL 语句实现对 Competition 表进行创建，注意主码、外码表示，其中 Ano 为 9 字符,Gno 为 4 字符, CGrade 为短整数,Crank 为短整数；

```
CREATE TABLE Competition
```

```
( Ano char (9),
```

```
  Gno char (4),
```

```
  CGrade smallint,
```

```
  Crank smallint,
```

```
    PRIMARY KEY (Ano, Gno),
```

```
    FOREIGN KEY (Ano) REFERENCES Athlete (Ano),
```

```
    FOREIGN KEY (Gno) REFERENCES Game (Gno))
```

6. 用 SQL 语句查询运动员编号为“2019005001”的运动员参加比赛项目获得名次最高的那项比赛的比赛项目名称。

```
SELECT Gname FROM Game
```

```
WHERE Gno in (SELECT Gno FROM Competition
```

```
WHERE Ano='2019005001' and Crank<=(SELECT min(Crank)
```

```
FROM Competition WHERE Ano='2019005001'))
```

7. 用 SQL 语句查询参加了 3 个以上比赛项目（不含 3 门）的运动员编号。

```
SELECT Ano FROM Competition  
GROUP BY Ano HAVING COUNT(*)>3
```

8. 用 SQL 语句将比赛项目编号为“7”的项目名称修改为“羽毛球混双”。

```
UPDATE Game SET Gname = '羽毛球混双' WHERE Gno = '7'
```

9. 创建一个惠州学院（hzu）运动员视图 A_hzu_VIEW，包括运动员编号、姓名、性别。

```
CREATE VIEW A_hzu_VIEW  
AS  
SELECT Ano,Aname,Asex FROM Athlete WHERE Sunit='hzu'
```

在工厂中，一个工人 (worker) 可以生产不同产品 (product)，一个产品也会有很多工人来生产，工人生产产品根据生产个数 (number) 获得工资，有如下关系表用于描述工人生产产品情况，关系中有下划线的表示主码。

Worker (Wno, Wname, Wsex, Wage, Wdept)

工人表：表示工号、姓名、性别、年龄、所在部门

Product (Pno, Pname, Pprice)

产品表：表示产品号、产品名、价格

WP (Wno, Pno, number)

工人生产表：工号、产品号、生产个数

请用关系代数及 SQL 语句完成如下题目：

1. 用关系代数实现生产了 1 号产品的工人工号。

$\pi_{Wno}(\sigma_{Pno='1'}(WP))$

2. 用关系代数实现部门为一车间 (workshop one) 的工人生产情况。

$\pi_{Wno}(\sigma_{Wdept='workshop one'}(Worker)) \bowtie WP$

3. 用关系代数实现生产 1 号产品并且个数在 150 个以上(不含 150)的工人工号和姓名。

$\pi_{Wno}(\sigma_{Pno='1' \wedge number > 150}(WP)) \bowtie \pi_{Wno, Wname}(Worker)$

4. 用 SQL 语句实现生产 1 号产品并且个数在 150 个以上(不含 150)的工人工号和姓名。

Select Worker.Wno,Wname

From Worker,WP

Where Worker.Wno=WP.Wno and WP.Pno='1' and WP.number>150

5. 用 SQL 语句实现对 WP 表的创建，注意主码、外码表示，其中 Wno 为 4 字符,Pno 为 4 字符, number 为短整数；

CREATE TABLE WP

(
 Wno char (4),
 Pno char (4),
 number smallint,
 PRIMARY KEY (Wno, Pno),
 FOREIGN KEY (Wno) REFERENCES Worker (Wno),
 FOREIGN KEY (Pno) REFERENCES Product (Pno)
)

6. 用 SQL 语句查询工人工号为“2207”的生产产品的总个数。

SELECT SUM(number)

FROM WP

WHERE Wno='2207'

7. 用 SQL 语句查询生产了 3 种以上 (不含 3 门) 产品的工人工号。

SELECT Wno

```
FROM WP  
GROUP BY Wno  
HAVING COUNT(*)>3
```

8. 用 SQL 语句将产品号为“0077”的产品名称修改为“4 分六角螺丝”。

```
UPDATE Product  
SET Pname ='4 分六角螺丝'  
WHERE Pno ='0077'
```

9. 为 WP 表创建一个索引（名为 WP_index），工号升序，产品号降序。

```
CREATE INDEX WP_index on WP (Wno ASC,Pno DESC)
```

有如下关系表用于描述产品的销售情况，关系中有下划线的表示主码。

Product (P_ID, Pname, Pstyle, price)

产品表： 用于表示产品号、产品名、规格和单价

Store (S_ID, Sname, Slocation)

仓库表： 用于表示仓库号、仓库名和仓库位置

PS (P_ID, S_ID, quantity)

库存表： 表示产品号、仓库号和库存数量， P_ID, S_ID 分别参照 Product: P_ID, Store: S_ID

Order (O_ID, customer, time)

订单表： 表示订单号、客户和下单时间

Order_Item(O_ID, line, P_ID, quantity, total_price)

订单项表： 表示订单号、行号、产品名、所定数量和小计金额

请用关系代数及 SQL 语句完成如下题目：

1. 用关系代数实现规格为“Cloth”的产品名称

$\pi_{Pname} \sigma_{Pstyle = 'Cloth'} (Product)$

2. 用 SQL 语句实现对 PS 表的创建，注意主码、外码表示，其中 P_ID 为 5 字符, S_ID 为 2 字符, quantity 为整数；

CREATE TABLE PS

```
(  
    P_ID char (5),  
    S_ID char (2),  
    Quantity int,  
    PRIMARY KEY (P_ID, S_ID),  
    FOREIGN KEY (P_ID) REFERENCES Product (P_ID),  
    FOREIGN KEY (S_ID) REFERENCES Store (S_ID)  
)
```

3. 用 SQL 语句查询库存数量超过 50 的产品 ID

SELECT P_ID

FROM PS, Store, Product

WHERE Product.P_ID=PS.P_ID and Store.S_ID=PS.S_ID and quantity>50

4. 用 SQL 语句查询客户“李辉”的所有订单及详细订单项情况

SELECT *

FROM Order, Order_Item

WHERE Order.O_ID = Order_Item.O_ID and Order.customer='李辉'

5. 用 SQL 语句在 Store 表中将仓库名为“成衣仓”改为“成品仓”

UPDATE Store

SET Sname = '成品仓'

WHERE Sname = '成衣仓'

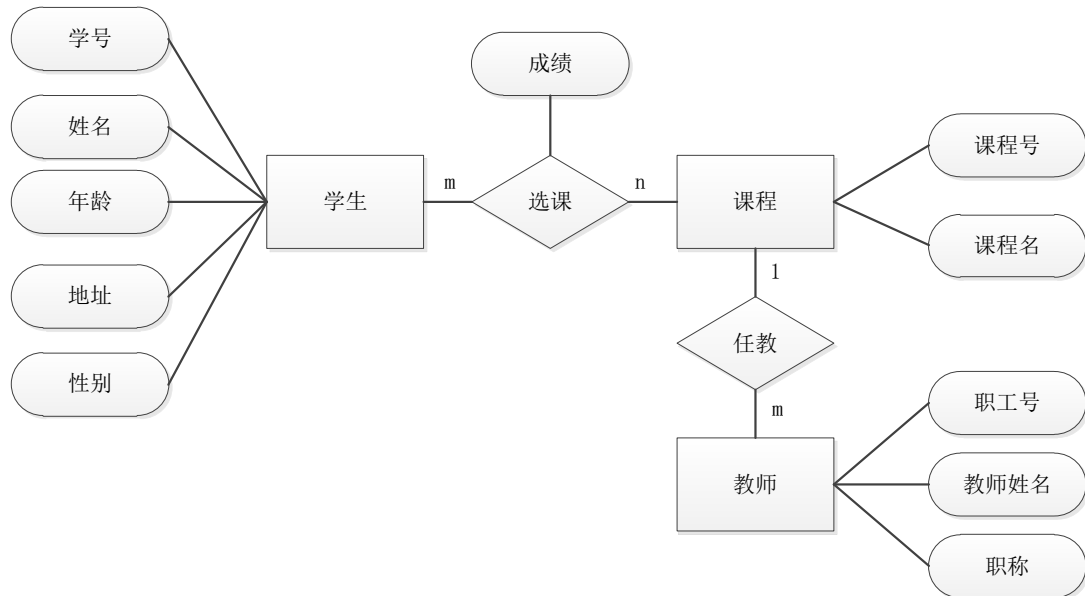
6. 用 SQL 语句为 Order_Item 表中 quantity 按升序建立名为 QQ 的索引

```
CREATE INDEX QQ ON Order_Item(quantity)
```

设计分析题

假设每个学生选修若干门课程，且每个学生每选一门课只有一个成绩，每个教师只担任一门课的教学，一门课由若干教师任教。“学生”有属性：学号、姓名、地址、年龄、性别。“教师”有属性：职工号、教师姓名、职称，“课程”有属性：课程号、课程名。

(1) 试画出 ER 图，并注明属性和联系类型。



(2) 将 E-R 图转换成关系模型，并注明主码和外码

学生 (学号, 姓名, 地址, 年龄, 性别)

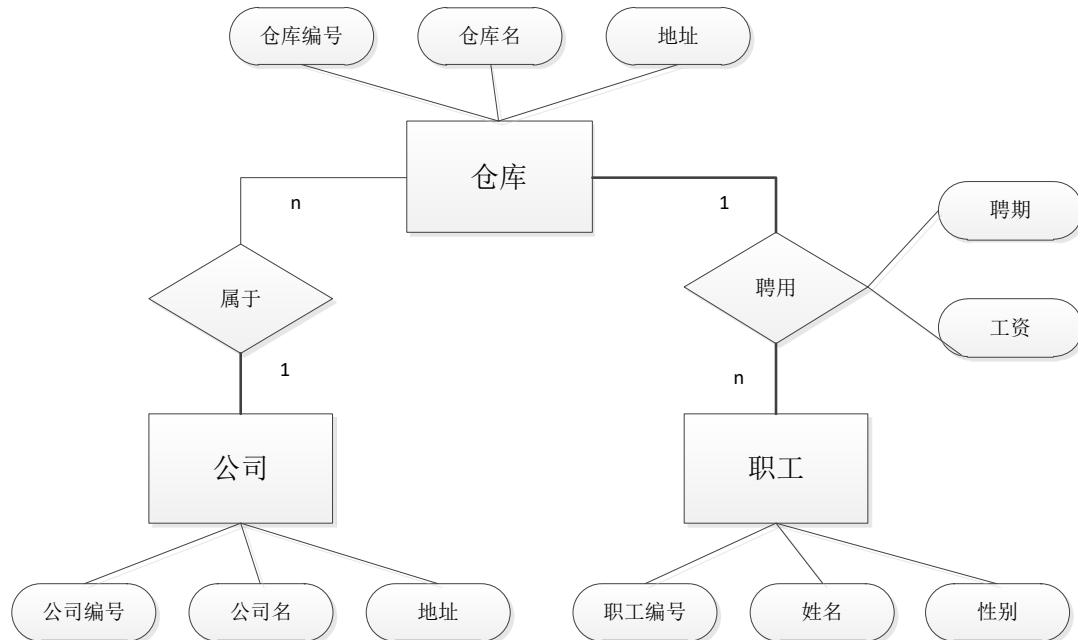
课程 (课程号, 课程名)

教师 (职工号, 教师姓名, 职称, **课程**)

选课 (学号, 课程号, 成绩)

设某商业公司数据库中有三个实体集，一是“公司”实体集，属性有公司编号、公司名、地址等；二是“仓库”实体集，属性有仓库编号、仓库名、地址等；三是“职工”实体集，属性有职工编号、姓名、性别等。每个公司有若干个仓库，每个仓库只能属于1个公司，每个仓库可聘用若干职工，每个职工只能在一个仓库工作，仓库聘用职工有聘期和工资。

(1) 试画出 E-R 图



(2) 将 E-R 图转换成关系模型，并注明主码和外码

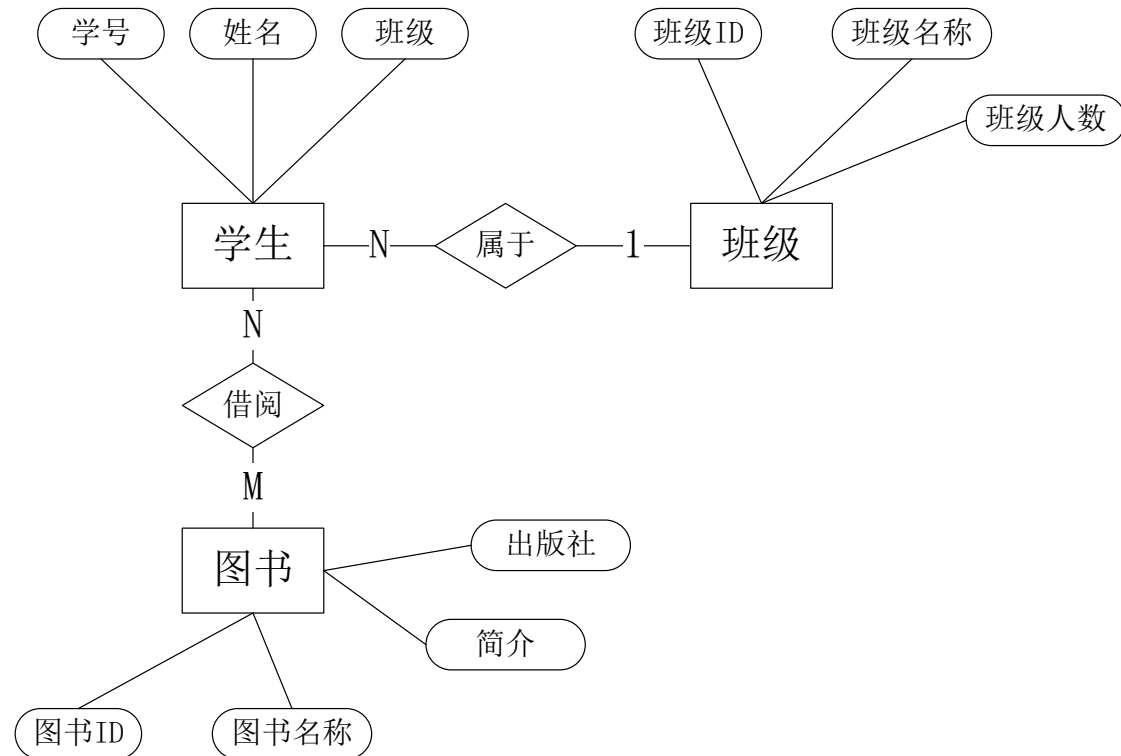
仓库 (仓库编号, 仓库名, 地址, 公司编号)

公司 (公司编号, 公司名, 地址)

职工 (职工编号, 姓名, 性别, 仓库编号, 聘期, 工资)

阅览室现有三个实体对象：学生、班级和图书。学生包括学号、姓名、班级等属性，一个学生只能属于一个班级；一个班级可以包括多个学生；一个学生可以借阅多本图书；一本图书可以被多个学生借阅。

(1) 自行设计三个对象属性，画出学生、班级、图书三个实体的 E-R 图，要求描述清楚各自属性、三个实体之间的联系；



(2) 写出所形成的关系模式，注意实体和参照完整性。

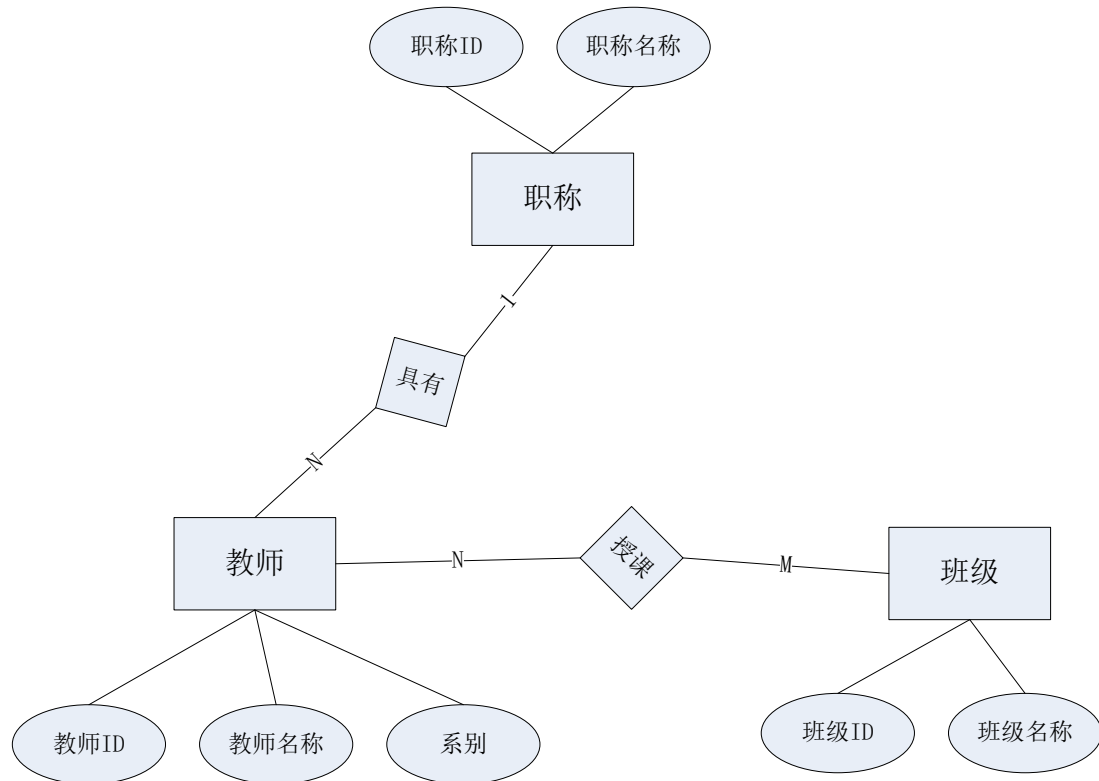
学生 (学号, 姓名, 班级 ID)

图书 (图书 ID, 图书名称, 简介, 出版社)

班级 (班级 ID, 班级名称, 班级人数)

现有三个实体对象：教师、职称和班级。职称是诸如教授、副教授、讲师、助教等说明教师具有的级别，一个教师只能具有一种职称；每个教师可以教授多个班级的课程；每个班级可以有多个教师教授。

(1) 请画出教师、职称、班级三个实体的 E-R 图，要求描述清楚各自属性、三个实体之间的联系；



(2) 写出所形成的关系模式，注意实体和参照完整性。

教师 (教师 ID, 教师姓名, 系别, 职称 ID)

班级 (班级 ID, 班级名称)

职称 (职称 ID, 职称名称)

证明

(1) 如果 R 的所有属性都是主属性，则 R 是 3NF。

关系模式 $R<U, F>$ 中若不存在这样的候选码 X ，属性组 Y 及非主属性 $Z(Z \notin Y)$ 使得 $X \rightarrow Y$ ， $Y \rightarrow Z$ 成立，则称 $R<U, F> \in 3NF$ 。因为如果 R 的所有属性都是主属性，即没有非主属性，则不存在非主属性对码的部分和传递函数依赖。因此如果 R 的所有属性都是主属性，则 R 是 3NF。

(2) 如果 R 的码包含 R 的所有属性（全码），则 R 是 BCNF。

BCNF 要求关系范式满足 1NF，且每一个函数依赖的决定因子都包含码。而全码的关系满足 1NF，且只有一个决定因子，即关系模式包含的所有属性。因此如果 R 的码包含 R 的所有属性（全码），则 R 是 BCNF。

(3) 若 $R \in BCNF$ ，则必有 $R \in 3NF$ 。

采用反证法

假设当 $R \in BCNF$ 时， $R \notin 3NF$

若 $R \in BCNF$ ，则有每一个函数依赖的决定因素必包含候选码

若 $R \notin 3NF$ ，则 R 属于 1NF 或者 2NF，若 $R \in 1NF$ ，则有非主属性对码的部分依赖

$(X, Y) \rightarrow Z$ ，且 $Y \rightarrow Z$ ，其中 $Y \subseteq (X, Y)$ ，而 Y 不包含码，和 $R \in BCNF$ 的定义相矛盾

若 $R \in 2NF$ ，则有非主属性对码的传递依赖：

$X \rightarrow Y$ ， $Y \not\rightarrow X$ ，且 $Y \rightarrow Z$ ($Z \notin Y$)

而 Y 不包含码（若 Y 是码，则有 $Y \rightarrow X$ ，就不是传递依赖了），和 $R \in BCNF$ 的定义相矛盾

因此若 $R \in BCNF$ ，则必有 $R \in 3NF$

(4) 若 $R \in 3NF$ ，则必有 $R \in 2NF$ 。

采用反证法

假设当 $R \in 3NF$ 时， $R \notin 2NF$

若 $R \in 3NF$ ，则不存在非主属性对码的传递依赖，即不存在：

$X \rightarrow Y$ ， $Y \not\rightarrow X$ ，且 $Y \rightarrow Z$ ($Z \notin Y$)

若 $R \notin 2NF$ ，则有 $(O, P) \rightarrow Q$ ， P 为 (O, P) 的真子集且 $P \rightarrow Q$ ，因此则有

$(O, P) \rightarrow P$ ， $P \not\rightarrow (O, P)$ ，且 $P \rightarrow Q$ ($Q \notin P$)

此处 (O, P) 即为上述所言之 X ， P 即是 Y ，而 Q 即是 Z ，存在非主属性 P 对码 (O, P) 传递依赖问题，与 $R \in 3NF$ 定义矛盾。

因此若 $R \in 3NF$ ，则必有 $R \in 2NF$