

作业答案

作业

- 计算机产品数据库，写出关系代数表达式：
 - 查询所有速度在2.0以上的产品（包括PC和笔记本）的型号和价格
 - $\pi_{\text{model, price}}(\sigma_{\text{speed} \geq 2.0}(\mathbf{PC})) \cup \pi_{\text{model, price}}(\sigma_{\text{speed} \geq 2.0}(\mathbf{Laptop}))$
 - 查询所有只卖笔记本而不卖PC的制造商。
 - $\pi_{\text{maker}}(\sigma_{\text{type}=\text{laptop}}(\mathbf{Product})) - \pi_{\text{maker}}(\sigma_{\text{type}=\text{PC}}(\mathbf{Product}))$

作业

- 计算机产品数据库，写出关系代数表达式（难点：重命名运算）：
 - 查询所有具有相同**CPU**速度和内存容量的**PC**型号对，每对只列出一次，如：列出了(i, j)就不要列出(j, i)。（提示：型号可以比较大小）
 - $\pi_{PC1.model, PC2.model}(\rho_{PC1}(PC) \bowtie_{PC1.speed=PC2.speed \wedge PC1.ram=PC2.ram \wedge PC1.model > PC2.model} \rho_{PC2}(PC))$
 - 查询最快速度的**PC** (提示：最快速度意味着不小于任何其它电脑的速度，可以从找出所有小于某一速度的角度解决此问题)
 - $\pi_{model}(PC) - \pi_{PC1.model}(\rho_{PC1}(PC) \bowtie_{PC1.speed < PC2.speed} \rho_{PC2}(PC))$

课堂练习：银行客户管理

分行表： *branch* (*branch-name*, *branch-city*, *assets*)

客户表： *customer* (*customer-name*, *customer-street*, *customer-city*)

存款账户表： *account* (*account-number*, *branch-name*, *balance*)

贷款账户表： *loan* (*loan-number*, *branch-name*, *amount*)

存款人表： *depositor* (*customer-name*, *account-number*)

贷款人表： *borrower* (*customer-name*, *loan-number*)

- *branch* (branch-name, branch-city, assets)
- *customer* (customer-name, customer-street, customer-city)
- *account* (account-number, branch-name, balance)
- *loan* (loan-number, branch-name, amount)
- *depositor* (customer-name, account-number)
- *borrower* (customer-name, loan-number)
- 找出在银行中同时有存款和贷款账户的客户（**分别用带IN和EXIST谓词的嵌套查询完成**）
- 找出那些总资产至少比位于**Brooklyn** 的某一家支行要多的支行名字（**用ANY谓词完成**）
- 找出平均存款余额最高的分行名字

- *depositor* (*customer-name*, *account-number*)
- *borrower* (*customer-name*, *loan-number*)
- 找出在银行中同时有存款和贷款账户的客户

```
select  distinct customer-name
from    borrower
where   customer-name in
        (select customer-name
         from   depositor)
```

```
select  distinct customer-name
from    borrower
where   exists
        (select  *
         from    depositor
         where   depositor.customer-name
                = borrower.customer-name)
```

■ *branch* (*branch-name*, *branch-city*, *assets*)

- 找出那些总资产至少比位于Brooklyn 的某一家支行要多的支行名字

```
select branch-name
from branch
where assets > any (select assets
                    from branch
                    where branch-city='Brooklyn')
```

■ *account* (*account-number*, *branch-name*, *balance*)

- 找出平均贷款余额最高的分行名字

```
select branch-name
from account
where balance = (select max(balance)
                  from account)
```


■ *account* (*account-number*, *branch-name*, *balance*)

- 找出平均贷款余额最高的分行名字

```
select  branch-name
from    account
group by branch-name
having  avg(balance) >= all(select  avg(balance)
                             from    account
                             group by branch-name)
```

作业

- 计算机产品数据库
 - Product(maker,model, type)
 - PC(model, speed, ram, hd, price)
 - Laptop(model, speed, ram, hd, screen, price)

Product

maker	model	type
A	1001	pc
A	1002	pc
A	2004	laptop
B	1003	pc
B	2005	laptop
C	1004	pc

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649

Laptop

model	speed	ram	hd	screen	price
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500

作业

- 计算机产品数据库，写出**SQL**查询语句：
 - 查询每个制造商及其生产的最低价格的笔记本型号
 - 查询生产的笔记本的硬盘容量不小于**100GB**的制造商
 - 查询生产最快速度的计算机的制造商

Product

maker	model	type
A	1001	pc
A	1002	pc
A	2004	laptop
B	1003	pc
B	2005	laptop
C	1004	pc

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649

Laptop

model	speed	ram	hd	screen	price
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500

- 查询每个制造商及其生产的最低价格的笔记本型号
 - 查询制造商maker生产的笔记本的最低价格

`select min(Laptop.price)`

`from Product, Laptop`

`where Product.model = Laptop.model and`

`Product.maker = maker`

查询每个制造商及其生产的最低价格的笔记本型号（难点：相关嵌套子查询）

```
select P1 maker, P1.model
from Product P1, Laptop L1
where P1.model = L1.model
      and L1.price = (select min(L2.price)
                      from Product P2, Laptop L2
                      where P2.model = L2.model
                        and P2.maker = P1.maker)
```

查询生产的笔记本的硬盘容量不
小于**100GB**的制造商

```
select distinct Product.maker  
from Product, Laptop  
where Product.model = Laptop.model and  
Laptop.hd >= 100
```

查询生产最快速度的PC的制造商

```
select DISTINCT P.maker
from Product P
where P.model in (
    select PC.model
    from PC
    where PC.speed = (
        select MAX(PC1.speed)
        from PC1
    )
)
```

Product

maker	model	type
A	1001	pc
A	1002	pc
A	2004	laptop
B	1003	pc
B	2005	laptop
C	1004	pc

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649

Laptop

model	speed	ram	hd	screen	price
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500

- 查询生产最快速度的计算机的制造商
 - 查询所有计算机（包括PC和笔记本）的型号和速度
- ```
select PC.model, PC.speed
from PC
UNION
select LP.model, LP.speed
from Laptop LP
```



# 查询生产最快速度的计算机的制造商

```
select DISTINCT P.maker
from Product P
where P.model in (
 select Computer.model
 from Computer
 where Computer.speed = (
 select MAX(Computer1.speed)
 from Computer1
)
)
```

# 查询生产最快速度的计算机的制造商

```
select DISTINCT P.maker
from Product P
where P.model in (
 select Computer.model
 from (select PC.model, PC.speed
 from PC
 UNION
 select LP.model, LP.speed
 from Laptop LP) AS Computer
 where Computer.speed = (
 select MAX(Computer1.speed)
 from (select PC1.model, PC1.speed
 from PC PC1
 UNION
 select Lp1.model, Lp1.speed
 from Laptop Lp1) AS Computer1)
)
```

# 作业

**Product**

| maker | model | type   |
|-------|-------|--------|
| A     | 1001  | pc     |
| A     | 1002  | pc     |
| A     | 2004  | laptop |
| B     | 1003  | pc     |
| B     | 2005  | laptop |
| C     | 1004  | pc     |

**PC**

| model | speed | ram  | hd  | price |
|-------|-------|------|-----|-------|
| 1001  | 2.66  | 1024 | 250 | 2114  |
| 1002  | 2.10  | 512  | 250 | 995   |
| 1003  | 1.42  | 512  | 80  | 478   |
| 1004  | 2.80  | 1024 | 250 | 649   |

**Laptop**

| model | speed | ram  | hd  | screen | price |
|-------|-------|------|-----|--------|-------|
| 2004  | 2.00  | 512  | 60  | 13.3   | 1150  |
| 2005  | 2.16  | 1024 | 120 | 17.0   | 2500  |

用SQL完成以下操作：

- 1、用**差集法**查询选修了全部课程的学生姓名。
- 2、将所有内存容量为**1024**的笔记本价格下调**200**元。
- 3、创建制造商**B**生产的所有产品的型号和价格信息的视图。

1、用**差集法**查询选修了全部课程的学生姓名（难点：exists和嵌套子查询）

```
select Sname
from Student S
where not exists
 ((select Cno
 from Course)
 except
 (select SC.Cno
 from Student S1, SC
 where S1.Sno = SC.Sno and
 S1.Sname = S.Same))
```

## 2、将所有内存容量为1024的笔记本价格下调200元

update Laptop

set price=price-200

where ram=1024

### 3、创建制造商B生产的所有产品的型号和价格信息的视图。

```
create view B-Computers
```

```
as
```

```
select model, price
```

```
from PC
```

```
where model in
```

```
 (select model
```

```
 from Product
```

```
 where maker = B)
```

```
union
```

```
select model, price
```

```
from Laptop
```

```
where model in
```

```
 (select model
```

```
 from Product
```

```
 where maker = B)
```

### 3、创建制造商**B**生产的所有产品的型号和价格信息的视图（难点：集合查询和连接查询的组合）

```
create view B-Computers
as
select PC.model, PC.price
from Product P, PC
where P.maker='B' and P.model = PC.model
union
select L.model, L.price
from Product P, Laptop L
where P.maker='B' and P.model = L.model
```

# 作业

作业：习题6.2

出现问题：

- 没有注意语义中“学生参加某学会有一个入会年份”，导致没有写出关系模式“学生—学会SP”。
- 没有注意到(专业名, 入学年份)→班号是班级C的候选码。



## 关系模式:

学生S(学号, 姓名, 出生年月, 系名, 班号, 宿舍区)

班级C(班号, 专业名, 系名, 班级人数, 入学年份)

系D(系号, 系名, 系办公地点, 系人数)

学会P(学会名, 成立年份, 地点, 会员人数)

学生—学会SP(学号, 学会名, 入会年份)

## 极小函数依赖集

S: 学号 $\rightarrow$ 姓名, 学号 $\rightarrow$ 出生年月, 学号 $\rightarrow$ 班号, 班号 $\rightarrow$ 系名, 系名 $\rightarrow$ 宿舍区

C: 班号 $\rightarrow$ 专业名, 班号 $\rightarrow$ 班级人数, 班号 $\rightarrow$ 入学年份, 专业名 $\rightarrow$ 系名, (专业名, 入学年份) $\rightarrow$ 班号 /\*每个专业每年只招收一个班\*/

D: 系号 $\rightarrow$ 系名, 系名 $\rightarrow$ 系号, 系号 $\rightarrow$ 系办公地点, 系号 $\rightarrow$ 系人数

P: 学会名 $\rightarrow$ 成立年份, 学会名 $\rightarrow$ 地点, 学会名 $\rightarrow$ 会员人数

SP: (学号, 学会名) $\rightarrow$ 入会年份 /\*学生参加某学会要有入会年份\*/

## 关系模式:

学生S(学号, 姓名, 出生年月, 系名, 班号, 宿舍区)

班级C(班号, 专业名, 系名, 班级人数, 入学年份)

系D(系号, 系名, 系办公地点, 系人数)

学会P(学会名, 成立年份, 地点, 会员人数)

学生—学会SP(学号, 学会名, 入会年份)

## S中存在的数据依赖:

学号 $\rightarrow$ 班号, 班号 $\rightarrow$ 系名, 系名 $\rightarrow$ 宿舍区

传递函数依赖: 学号 $\rightarrow$ 系名; 班号 $\rightarrow$ 宿舍区; 学号 $\rightarrow$ 宿舍区

## C中存在的数据依赖:

班号 $\rightarrow$ 专业名, 专业名 $\rightarrow$ 系名,

传递函数依赖: 班号 $\rightarrow$ 系名。

## 函数依赖左部是多属性的情况:

(学号, 学会名) $\rightarrow$ 入会年份, (专业名, 入学年份) $\rightarrow$ 班号

指出个关系的候选码，外部码，有没有全码。

| 关系 | 候选码                | 外部码     | 全码 |
|----|--------------------|---------|----|
| S  | 学号                 | 班号, 系名  | 无  |
| C  | 班号和<br>(专业名, 入学年份) | 系名      | 无  |
| D  | 系号和系名              | 无       | 无  |
| P  | 学会名                | 无       | 无  |
| SP | (学号, 学会名)          | 学号, 学会名 | 无  |

# 练习

针对QQ应用，设计概念模式和逻辑模式

提示 应包含如下信息（但不限于）：

用户

群

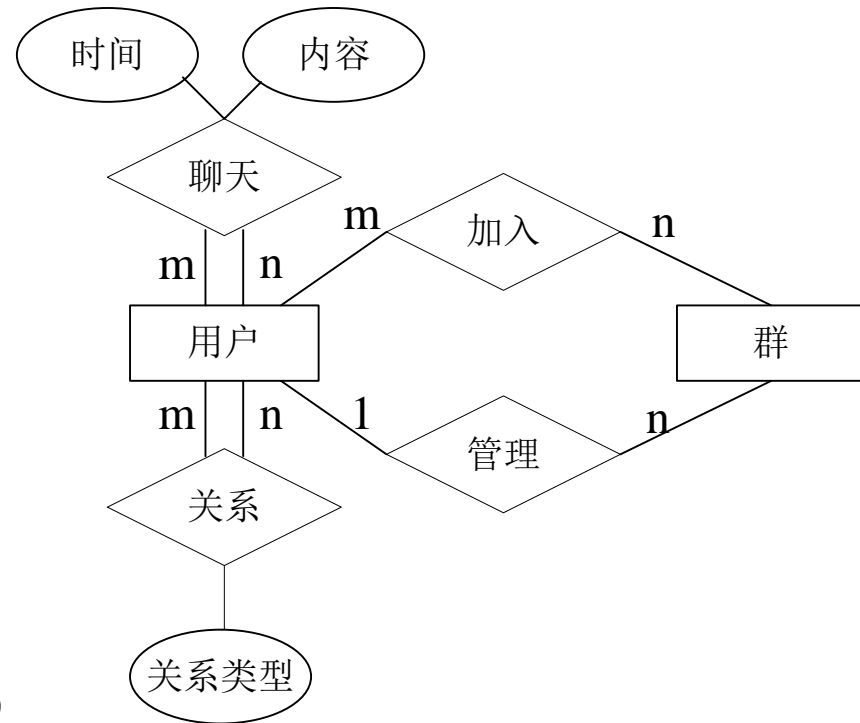
聊天记录

# QQ

- 实体
  - 用户
  - 群
- 联系
  - 关系（用户-用户）
  - 聊天（用户-用户）
  - 加入（用户-群）
  - 管理（用户-群）

# QQ

- 用户 (qq号, 昵称, .....)
- 群 (群号, 群名称, 管理员qq号, .....)
- 关系 (用户A qq号, 用户B qq号, 关系类型, .....)
- 聊天 (发送用户qq号, 接受用户qq号, 发送时间, 消息内容, .....)
- 加入 (用户qq号, 群号, .....)

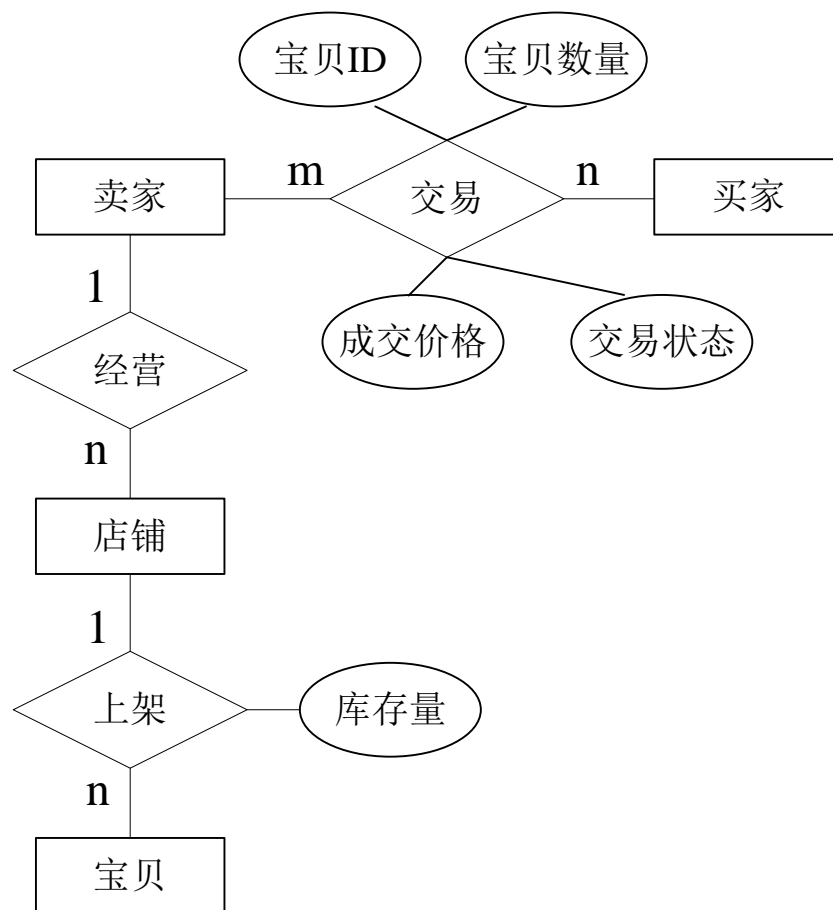


# 作业：淘宝网数据库设计

- 实体
  - 卖家
  - 买家
  - 店铺
  - 宝贝
- 联系
  - 经营（卖家-店铺）
  - 上架（店铺-宝贝）
  - 交易（卖家-买家）

# 淘宝网

- 卖家（卖家ID，卖家昵称，评级，.....）
- 店铺（店铺ID，店铺名，卖家ID，.....）
- 宝贝（宝贝ID，宝贝名，店铺ID，库存量，描述，成交记录，.....）
- 买家（买家ID，买家昵称，.....）
- 交易（卖家ID，买家ID，下单时间，宝贝ID，宝贝数量，成交价格，交易状态，.....）



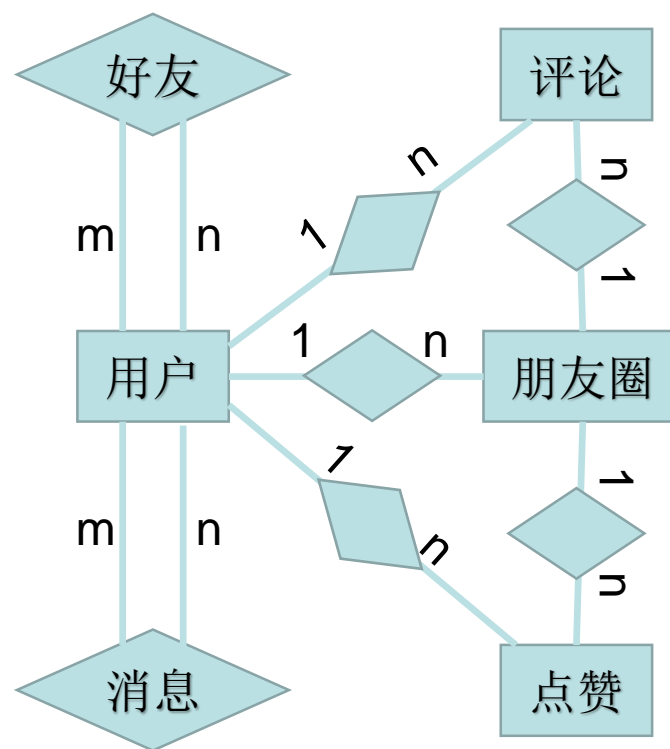


# 作业：微信数据库设计

- 实体
  - 用户
  - 朋友圈
  - 评论
  - 点赞
- 联系
  - 好友（用户-用户）
  - 消息（用户-用户）
  - 发朋友圈（用户-朋友圈）
  - 评论（用户-朋友圈）
  - 点赞（用户-朋友圈）

# 微信

- 用户（微信号、昵称、未读消息数、未读评论数、未读点赞数、.....）
- 好友（己方微信号、对方微信号）
- 消息（发送方微信号、接收方微信号、发送时间、消息内容）
- 朋友圈（朋友圈ID、发布者微信号、发布时间、内容、评论数、点赞数、.....）
- 评论（评论ID、评论者微信号、朋友圈ID、评论时间、评论内容、.....）
- 点赞（点赞ID、点赞者微信号、朋友圈ID）



# 微信

- 用户（微信号、昵称、未读消息数、未读评论数、未读点赞数、.....）
- 好友（己方微信号、对方微信号）
- 消息（发送方微信号、接收方微信号、发送时间、消息内容）
- 朋友圈（朋友圈ID、发布者微信号、发布时间、内容、评论数、点赞数、.....）
- 评论（评论者微信号、朋友圈ID、评论时间、评论内容、.....）
- 点赞（点赞者微信号、朋友圈ID）

