

# 数据库第一次作业

软件2203班 丁海桐 202226010304

2-1. 对图2-4 所示的大学教务管理数据库中的5个表，补充教师开课表、教室表课表。现列出这8个表的表名及其包含的字段名（即其模式）如下：

学院表：Department(deptNo, name, dean, address, phone)  
学生表：Student(studentNo, name, sex, birthday, nation, classNo, phone, familyAddr, deptro)  
课程表：Course(courseNo, name, hours, credit, book, semester, deptNo)  
教师表：Teacher(teacherno, name, rank, birthday, salary, cmail, phone, deptro)  
教师开课表：Teach(courseNo, semester, classNo, teacherno)  
学生选课表：Enroll(studentNo, courseNo, semester, classNo, teacherno, score)  
教室表：Classroom (ClassroomNo, type, seat num)  
排课表：Dispatch(courseNo, semester, classNo, period, weekday, turns, interval, classroomNo)

其中教师开课表的含义是：每门课可能有多个教学班，教学班的字段名为 classNo，每个教必学班要安排一位教师负责上课。教室表很好理解，classroomNo 是教室的号码，type 是教室的类型，seat num 是座位数。对于排课表，其数据实例如下：

回答下列问题：

1) 标识出每个表的主键：分析外键，如果有，则标识出来，指明它引用了哪个表的主键。

- Department表的主键是deptNo。(dean是否为外键存疑，P52页中说dean引用Teacher图的主键TeacherNo，但是课本P29页图2-4中dean的值为人名，TeacherNo值为一串数字，所以这里不把其当做外键)
- Student表的主键是studentNo。
- Course表的主键是courseNo，deptNo是外键，引用了Department表的主键deptNo。
- Teacher表的主键是teacherNo，deptNo是外键，引用了Department表的主键deptNo。
- Teach表的主键是(courseNo, semester, classNo)，courseNo是外键，引用了Course表的主键courseNo，teacherNo是外键，引用了Teacher表的主键teacherNo。
- Enroll表的主键是(studentNo, courseNo, semester)，studentNo、courseNo和teacherNo是外键，分别引用了Student表、Course表和Teacher表的主键。
- Classroom表的主键是ClassroomNo。
- Dispatch表的主键是(courseNo, semester, classNo, period, weekday, turns, interval)。courseNo和classroomNo是外键，分别引用了Course表和Classroom表的主键。

2) 课程表中的semester是指在第几个学期选修。本科4年，共有8个学期。字段credit 是学分，字段hours 是学时。请对课程表中的credit字段以及教师表中的rank、salary 这两个字段，给出其域约束（数据类型，取值范围）。

- credit字段的数据类型为SHORTINT，取值范围应该是[0, 32767]，代表课程的学分数。
- rank字段的数据类型为VARCHAR，取值范围可能是教师的职称，如"助教"、"讲师"、"副教授"等。
- salary字段的数据类型为FLAOT，取值范围应该是大于等于0的实数，代表教师的薪水。

3) 使用关系代数写出表达式，用来生成如下用户所需业务数据：

- 列出所有在2000 年出生的女生，输出项包括学号、姓名、出生日期。

$$\pi_{\text{studentNo, name, birthday}}(\sigma_{\text{sex}='女' \text{ AND birthday like '2000\%'}}(Student))$$

- 学生张珊，其学号为 200843407，请列出其已经选修的课程，输出项包括课程编号、课程名称、选修时间、学分、成绩。

$$\pi_{\text{courseNo, name, semester, credit, score}}(\sigma_{\text{studentNo}='200843407'}(Enroll \bowtie Course))$$

- 求出在 2011-1 学期选修了操作系统这门课(课程编号为 H161030009) 的学生名单，输出项包括学号、姓名、性别。

$$\pi_{\text{studentNo, name, sex}}(\sigma_{\text{semester}='2011-1' \text{ AND courseNo}='H161030009'}(Enroll \bowtie Student))$$

2-2. 某酒店集团公司在全国各城市开有酒店。每个酒店都有客房，客户可在网上先登录，然后预订，也可到店时现场预订。其住宿业务数据库中有如下4个表：

Hotel(hotelNo, name, city, address, phone)  
Room(roomNo, hotelNo, type, price)  
Booking (hotelNo, roomNo, guestNo, dateFrom, dateTo)  
Guest(guestNo, password, name, email, phone)

其中房间的类型有单人间、双人间、商务间、豪华间。

1) 标识出4个表的主键。对有外键的，标识出它的外键。

- Hotel表的主键是hotelNo。
- Room表的主键是roomNo，hotelNo是外键，关联Hotel表的主键hotelNo。(这里认为此表中存储着全国所有该酒店的所有与房间，给每一个room一个像身份证一样的ID)

- Booking表的主键是(hotelNo, roomNo, guestNo, dateFrom), hotelNo、roomNo和guestNo都是外键, 分别关联Hotel表、Room表和Guest表的主键。
- Guest表的主键是guestNo。

## 2) 基于酒店住宿业务常识, 写出两个企业业务规则约束。

- 同一时间同一间房不能被两个用户同时预约。
- 房间的类型只有有单人间、双人间、商务间、豪华间。

## 3) 使用关系代数写出表达式, 为用户生成如下所需的业务数据:

- 对价格大于1500元的房间, 列出房间类型、其酒店名称及其所在城市。

$$\pi_{\text{Room.type, Hotel.name, Hotel.city}}(\sigma_{\text{Room.price} > 1500}(\text{Room} \bowtie \text{Hotel}))$$

- 某个客户现在想在位于长沙这个城市的酒店预订一个'单人间'房, 住宿期为 '2019-09-27' 一天, 请列出当前可预订房间的酒店名称、房间号、价格。

$$\pi_{\text{Hotel.name, Room.roomNo, Room.price}} \left( \sigma_{\substack{\text{Hotel.city} = \text{'长沙'} \\ \text{Room.type} = \text{'单人间'}}} (\text{Booking} \bowtie \text{Room} \bowtie \text{Hotel}) \right) \\ - \pi_{\text{Hotel.name, Room.roomNo, Room.price}} \left( \sigma_{\substack{\text{Hotel.city} = \text{'长沙'} \\ \text{Room.type} = \text{'单人间'} \\ \text{Booking.dateFrom} \leq \text{'2019-09-27'} \\ \text{Booking.dateTo} \geq \text{'2019-09-27'}}} (\text{Booking} \bowtie \text{Room} \bowtie \text{Hotel}) \right)$$

3-1 . 其酒店集团公司在全国各城市开设有酒店。每个酒店都有客房, 客户可在网上先登录, 然后预订, 也可到店时现场订房。其佳业分数协定中有如下4个表:

Hotel (hotelNo, name, city, address, phone)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, roomNo, guestNo, dateFrom, dateTo)

Guest (guestNo, password, name, city, email, phone, discount, creditNo)

其中Room表中房间类型type字段的取值有单人间、双人间、商务间、家庭间; price是指住宿一天的房价。客户表Guest中, city字段是指客户的户籍地; discount是客户能享受的折扣, 即优惠; creditNo是信用卡号, 用于住宿后收取房费。

用 SQL 表示如下22个数据操作:

注意1: 这里我定义Room表的主键为roomNo

注意2: 预定双人间只会产生一个订单, 只涉及一个Guest

## 1) !!!列出位于长沙这个城市的所有酒店的基本情况, 包括酒店名称、房间类型、房间数量, 按酒店名称排序。

```
SELECT h.name AS 酒店名称, r.type AS 房间类型, COUNT(r.roomNo) AS 房间数量
FROM Hotel AS h, Room AS r
WHERE h.hotelNo = r.hotelNo
      AND h.city = '长沙'
GROUP BY h.name, r.type
ORDER BY h.name;
```

## 2) 列出2019-09-29这天, 住在位于长沙的该酒店的所有客人的姓名、所住房间号, 按姓名排序。

```
SELECT g.name AS 客人姓名, b.roomNo AS 房间号
FROM Booking AS b, Guest AS g, hotel AS h
WHERE b.dateFrom <= '2019-09-29'
# WHERE '2019-09-29' BETWEEN b.dateFrom AND b.dateTo
      AND b.dateTo >= '2019-09-29'
      AND b.guestNo = g.guestNo
      AND h.city = '长沙'
ORDER BY g.name;
```

## 3) 列出位于长沙的酒店, 房价在140元以下的所有双人间, 按酒店名称、价格升序排序。

```
SELECT h.name AS 酒店名称, r.price AS 价格, h.address AS 地址, h.phone AS 前台电话
FROM hotel AS h, room AS r
WHERE h.hotelNo = r.hotelNo
      AND h.city = '长沙'
      AND r.price < 140
      AND r.type = '双人床'
ORDER BY h.name, r.price
```

**4) 2019年8月, 有多少不同的客人订房?**

```
SELECT COUNT(distinct guestNo) AS 八月顾客数
FROM booking AS b
WHERE b.dateFrom <= '2019-08-01'
      AND b.dateTo <= '2019-08-31'
```

**5) 列出dateTo字段值为NULL (即还没有确定) 的预订记录。**

```
SELECT *
FROM booking
WHERE booking.dateTo IS NULL
```

**6) 集团公司共有多少酒店?**

```
SELECT COUNT(*) AS 酒店数量
FROM hotel
```

**7) 以城市分组, 求出其酒店的房间平均价格是多少。**

```
SELECT h.city AS 城市, AVG(r.price) AS 房间平均价格
FROM room AS r, hotel AS h
WHERE r.hotelNo = h.hotelNo
GROUP BY h.city
```

**8) 列出该酒店在不同城市, 其包含的房间类型及其平均价格。输出所在城市名称、房间类型、平均价格。**

```
SELECT h.city AS 城市, r.type AS 房间类型, AVG(r.price) AS 房间平均价格
FROM room AS r, hotel AS h
WHERE r.hotelNo = h.hotelNo
GROUP BY h.city, r.type;
```

**9) 列出在2019-09-29这天, 户籍地为北京的住客情况, 包含酒店所在城市、酒店名称、房间号、姓名、客户编号。**

```
SELECT h.city AS 酒店所在城市, h.name AS 酒店名称, b.roomNo AS 房间号,
g.name AS 姓名, g.guestNo AS 客户编号
FROM booking AS b, guest AS g, hotel h
WHERE b.dateFrom <= '2019-09-29'
      AND b.dateTo >= '2019-09-29'
      AND g.city = '北京'
      AND h.hotelNo = b.hotelNo
      AND g.guestNo = b.guestNo
```

**10) 对于位于长沙的该酒店, 假定它的所有房间都住满了, 那么它一天的总收入是多少? 不考虑客户的折扣。**

```
SELECT SUM(r.price)
FROM hotel AS h, room AS r
WHERE h.hotelNo = r.hotelNo
      AND h.city = '长沙';
```

**11) 昨天 (2019-09-29) 位于长沙的该酒店, 因房间有客人入住了, 其实际总收入是多少? 不考虑客户的折扣。**

```
SELECT SUM(r.price) AS 实际总收入
FROM hotel AS h, room AS r, booking AS b
WHERE b.hotelNo = h.hotelNo
      AND b.roomNo = r.roomNo
      AND h.city = '长沙'
      AND '2019-09-29' BETWEEN b.dateFrom AND b.dateTo;
```

**12) !!!明天 (2019-09-30) 位于长沙的该酒店, 哪些房间空闲无人住? (题中提到了该酒店, 所以这里理解为该酒店在长沙只有一座)**

```
SELECT roomNo AS 空闲房间号
FROM Room
WHERE hotelNo IN (
    SELECT hotelNo
    FROM Hotel
    WHERE city = '长沙')
EXCEPT
SELECT roomNo
FROM Booking
WHERE dateFrom <= '2019-09-30' AND dateTo >= '2019-09-30';
```

13) !!!昨天 (2019-09-29) , 每个酒店的房间入住率是多少? 输出字段为城市名称、酒店名称、入住率。

```
SELECT h.city AS 城市名称,
       h.name AS 酒店名称,
       (SELECT COUNT(*) FROM Booking b WHERE b.hotelNo = h.hotelNo AND b.dateFrom <= '2019-09-29' AND b.dateTo >= '2019-09-29') *
       100.0 /
       (SELECT COUNT(*) FROM Room r WHERE r.hotelNo = h.hotelNo) AS 入住率
FROM Hotel h;
```

14) !!!今天 (2019-09-30) , 一位客户想在2019-10-01至2019-10-03入住位于长沙的该酒店, 他想预订一个双人间, 请为其列出所有可预订的双人间的房间号、价格。

```
SELECT r.roomNo, r.price
FROM room AS r
WHERE r.hotelNo IN (
    SELECT h.hotelNo
    FROM hotel AS h
    WHERE h.city = '长沙'
)
AND r.type = '双人间'
AND r.roomNo NOT IN (
    SELECT b.roomNo
    FROM booking AS b
    WHERE b.dateFrom <= '2019-10-03' AND b.dateTo >= '2019-10-01'
)
```

15) 昨天 (2019-09-29) , 对位于长沙的酒店, 最常订的房间分别是哪种类型? (是指入住的房间中, 哪种类型的房间数最多。)

```
SELECT r.type AS 房间类型, COUNT(*) 预定数
FROM booking AS b, hotel AS h, room AS r
WHERE b.roomNo = r.roomNo
    AND b.hotelNo = h.hotelNo
    AND h.city = '长沙'
    AND dateFrom <= '2019-09-30' AND dateTo >= '2019-09-30'
GROUP BY r.type
ORDER BY COUNT(*) DESC LIMIT 1;
```

16) 请列出2019年8月, 预订酒店次数大于5次的客户, 包括姓名、身份证号、户籍地。 (这里理解为预约住的第一天dateFrom在八月份就可以)

```
SELECT g.name, g.creditNo, g.city
FROM guest AS g, booking AS b
WHERE b.dateFrom >= '2019-08-01'
    AND b.dateFrom <= '2019-08-31'
    AND b.guestNo = g.guestNo
GROUP BY g.guestNo
HAVING COUNT(*) > 5;
```

17) 向每个表中插入一行数据。

```
INSERT INTO hotel (hotelNo, name, city, address, phone)
VALUES ('H001', 'Sample Hotel', 'City', '123 Sample Address', '1234567890');

INSERT INTO room (roomNo, hotelNo, type, price)
VALUES ('R001', 'H001', 'Single Room', 100.00);

INSERT INTO booking (hotelNo, roomNo, guestNo, dateFrom, dateTo)
VALUES ('H001', 'R001', 'G001', '2024-03-19', '2024-03-21');

INSERT INTO guest (guestNo, password, name, city, email, phone, discount, creditNo)
VALUES ('G001', 'password123', 'John Doe', 'City', 'john.doe@example.com', '9876543210', 0.1, '1234-5678-9012');
```

18) 将所有房间的价格提高5%。

```
UPDATE room AS r
SET r.price = r.price * 1.05;
```

19) 创建与表Booking具有相同结构的表archival\_booking, 用于保存历史记录。用INSERT语句, 将表Booking中2019年1月1日前的预订数据转移到表archival\_booking中, 在表Booking中不再保留。

```
-- 创建 archival_booking 表, 结构与 booking 表相同
CREATE TABLE archival_booking LIKE booking;
```

```
-- 插入 Booking 表中2019年1月1日前的预订数据到 archival_booking 表中
INSERT INTO archival_booking
( SELECT *
FROM Booking
WHERE dateFrom < '2019-01-01');

-- 从 booking 表中删除2019年1月1日前的预订数据
DELETE FROM booking
WHERE dateFrom < '2019-01-01';
```

**20) 对于2019年预订酒店次数大于等于12次的客户，进行优惠，将其折扣调为原来折扣的90%。** (这里对于2019年预订酒店理解为: '2019-01-01' <= dateFrom <='2019-12-31')

```
UPDATE guest AS g
SET g.discount = g.discount * 0.9
WHERE g.guestNo IN
(
    SELECT guestNo
    FROM booking AS b
    WHERE dateFrom BETWEEN '2019-01-01' AND '2019-12-31'
    GROUP BY b.guestNo
    HAVING COUNT(*) >= 12
)
```

**21) 对自2017年以来，没有预订过酒店的客户，将其从客户表中删除。**

```
DELETE FROM guest
WHERE guestNo NOT IN
(
    SELECT distinct b.guestNo
    FROM booking AS b
    WHERE b.dateFrom >= '2017-01-01'
)
```

**22) 求出在2019年，对位于长沙的每个酒店都有过预订的客户，列出其姓名、身份证号、户籍地。**

```
-- 找出2019年预订过位于长沙的每个酒店的客户
SELECT b.guestNo
INTO changan_hotel_customers
FROM booking AS b, hotel AS h
WHERE h.city = '长沙'
    AND YEAR(b.dateFrom) = 2019
    AND b.hotelNo = h.hotelNo
GROUP BY b.guestNo
HAVING COUNT(DISTINCT b.hotelNo) = (
    SELECT COUNT(*)
    FROM Hotel
    WHERE city = '长沙'
);

-- 列出这些客户的姓名、身份证号、户籍地
SELECT g.name AS 姓名, g.creditNo AS 身份证号, g.city AS 户籍地
FROM Guest g
WHERE guestNo IN (
    SELECT guestNo
    FROM changan_hotel_customers
);
```