

## 主键约束：

Booking 中如果选择 **hotelNo, guestNo, roomNo** 为主键。分析：这就意味着一个客户预订某个旅馆的某个房间在整个一生中只能订一次；显然不符合实际情况。

如果选择 **hotelNo, guestNo, dateFrom** 为主键。分析：这就意味着一个客户某天在某个旅馆只能订一个房间；实际中，某个客户可能在某天在某个旅馆要订多个房间，例如在组织会议时，就有这种情况，因此不能满足业务要求。

如果选择 **hotelNo, roomNo, guestNo, dateFrom** 为主键。考虑到某个旅馆的某个房间在某天只可能被一个客户预订，因此不是最小属性集合。正确的主键选择应该是 **hotelNo, roomNo, dateFrom**。

如果选择 **hotelNo, roomNo, dateFrom, dateTo** 为主键。从语义上考虑，没有问题，能够确保行记录的唯一性。但是从逻辑上考虑，则有问题，因为它意味着可以出现 **hotelNo, roomNo, dateFrom** 字段值相同，但是 **dateTo** 字段值不相同的多行记录，这就违背了某一房间在某一天只能被预定一次的约束。

## 业务约束：

(a) 一个旅馆对应一个旅馆编号，没有重复的旅馆编号；评析：这应

该是主键约束，不是语义约束；

(b). 价格必须在 10 美元到 1000 美元之间。评析：这应该是域约束，不是语义约束；

(c) booking 表中的 hotelNo 字段的取值必须要在 Hotel 表中已存在。评析：这不是语义约束，而是外键约束；

(d). dateFrom 和 dateTo 必须大于今天日期；评析：这是语义约束。

问题：SQL 中如何表达该约束；

(e) 比如规定酒店的数目，超过该数目则不能添加该酒店。评析：不符合实际情况，不构成一条语义约束。

(f) 一个客户可以定多个房间，一个房间在一个时间只能有一个客户定；这是主键约束，不是语义约束；

(g) dateTo 必须大于或等于 dateFrom。这是语义约束。问题：SQL 中如何表达该约束；

因此，在考虑表的约束时，要依次按照主键约束，外键约束，域约束，语义约束来考虑，要有先后顺序，不能随意。主键约束对任何表都是必要的，要最先考虑的。然后是外键约束，再接下来是域约束，最后才是业务规则约束（即语义约束）。

(yjm) 某一旅馆的某一个房间在某一天的预定在 booking 表中不能有重叠。

//对于某个客户要在某个旅馆，定某个日期段（从 startDate 至 endDate）的某个类型的房间：列出可选的房间列表（hotelName, roomNo, type, price）；

写一个存储过程： Available\_room(String @ specified\_hotelName specified\_type, @ specified\_startDate, @ specified\_endDate)

```
SELECT @ specified_hotelName AS hotelName, roomNo, type ,  
price FROM room WHERE hotelNo = (SELECT hotelNo FROM  
hotel WHERE hotelName=@ specified_hotelName) AND  
type=@specified_type AND roomNo NOT IN (
```

```
SELECT hotelNo, roomNo INTO tempTbl FROM booking  
WHERE hotelNo = (SELECT hotelNo FROM hotel WHERE  
hotelName=@specified_hotelName) AND ((dateFrom>= @  
specified_startDate AND dateFrom <= @ specified_endDate) OR  
(dateTo>= @ specified_startDate AND dateTo <= @  
specified_endDate) OR (dateFrom < @ specified_startDate AND  
dateTo > @ specified_endDate))
```

```
)
```

在 **booking** 表中, 某一旅馆的某一个房间在某一天不能出现重复预定的情况。

某个客户想要在 **hotelNo** 为'01'的旅馆预定从'2012/4/10'至'2012/4/12'的一个'双人间'房间, 请列出他可选的房间列表 (**roomNo**)

```
SELECT roomNo FROM room WHERE hotelNo ='01' AND type='双人间' AND roomNo NOT IN (
```

```
    SELECT roomNo FROM booking WHERE hotelNo ='01' AND  
    ((dateFrom >= '2012/04/10' AND dateFrom <= '2012/04/12') OR  
    (dateTo >= '2012/04/10' AND dateTo <= '2012/04/12') OR  
    (dateFrom < '2012/04/10' AND dateTo > '2012/04/12'))  
)
```

3.22 今天是 2012/03/28, 问昨天每个旅馆空房损失分别是多少?

```
SELECT hotelname SUM(price) AS lossprice FROM  
booking,hotel,room WHERE booking.hotelno=hotel.hotelno
```

**=room.hotelno AND booking.roomno=room.roomno AND  
dateTo<'20120315' GROUP BY hotelname**

评析：1) 联接问题：必须按照表的内在关系来写等式，必须是：

(B.hotelNo=R.hotelNo AND B.roomNo=R.roomNo) AND R.hotelNo =

H.hotelNo。大家可以上机时比较上述联接与下面的联接

B.hotelNo=H.hotelNo AND B.roomNo=R.roomNo AND R.hotelNo =

H.hotelNo 间 结果的差异性。

2) 日期条件的判别：

应该是： dateFrom = today OR (dateFrom < today AND

dateTo>=today)

3) 得出的不是空房损失，而是收入，因此不符合要求；

4) 考虑会不会存在如下情况：某个旅馆的某个房间会出现多行记录。

现在是考虑预订，某个旅馆的某个房间在某天只可能存在一个预定，

因此不会有多行记录。

已被预定的房间：

**SELECT hotelname, R.hotelNo, R.roomNo, price FROM booking**

**AS B, hotel AS H, room AS R WHERE (B.hotelNo=R.hotelNo AND**

**B.roomNo=R.roomNo) AND R.hotelNo = H.hotelNo AND**

**dateFrom = today OR (dateFrom < today AND dateTo>=today);**

5) 知道了已被预定的房间，空房就是所有房间减去已被预定的房间；

```
SELECT hotelname, R.hotelNo, R.roomNo, price FROM hotel
AS H, room AS R WHERE R.hotelNo = H.hotelNo
```

**EXCEPT**

```
SELECT hotelname, R.hotelNo, R.roomNo, price FROM booking
AS B, hotel AS H, room AS R WHERE (B.hotelNo=R.hotelNo AND
B.roomNo=R.roomNo) AND R.hotelNo = H.hotelNo AND
dateFrom = today OR (dateFrom < today AND dateTo>=today)
```

6) 上面得出的是所有的空房清单，题目是要来求每个旅馆的空房损失，而不是每个空房间的损失，因此还要按照旅馆来进行分组统计；

```
SELECT hotelname, hotelNo, SUM (price) AS lostIncome FROM (
```

```
SELECT hotelname, R.hotelNo, R.roomNo, price FROM hotel
AS H, room AS R WHERE R.hotelNo = H.hotelNo
```

**EXCEPT**

```
SELECT hotelname, R.hotelNo, R.roomNo, price FROM booking
AS B, hotel AS H, room AS R WHERE (B.hotelNo=R.hotelNo
AND B.roomNo=R.roomNo) AND R.hotelNo = H.hotelNo AND
dateFrom = today OR (dateFrom < today AND dateTo>=today)
```

)

**GROUP BY** hotelname, hotelNo ;

**CREATE TABLE Booking (**

**hotelNo VARCHAR(4),**

**guestId VARCHAR(12) NOT NULL,**

**dateFrom DATE,**

**dateTo DATE NOTNULL,**

**roomNo CHAR(4),**

**PRIMARY(hotelNo, roomNo, dateFrom)**

FOREIGN KEY hotelNo, roomNo REFERENCES room(hotelNo, roomNo)

FOREIGN KEY guestId REFERENCES guest(guestId)

CHECK (dateTo >=dateFrom)

**}**