**实验2报告**

组员 龚仕奇，潘澳旋，刘欣宇

1. ER图

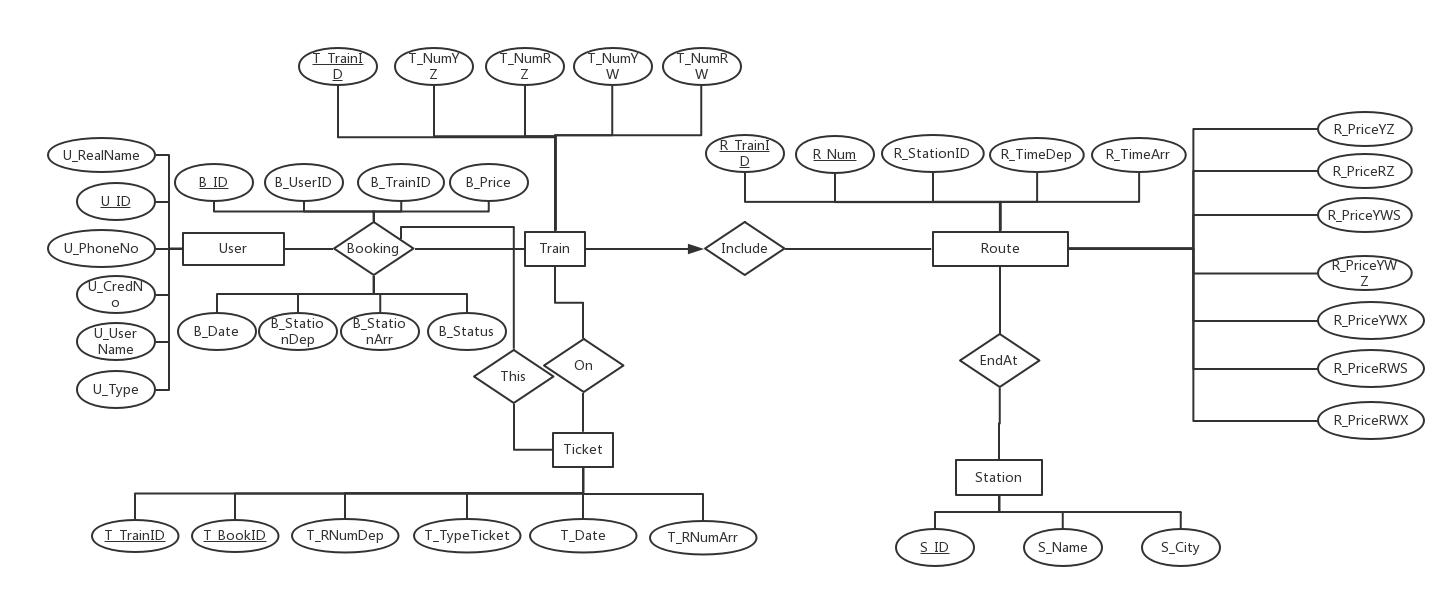


图 1 ER图

1. 关系模式

基本实现：

如ER图所示，我们的基本思路是将

以下为ER图实现为表后的表示：

1. 预订booking表：每条记录为一次订票信息，主键为b\_id，即唯一订单号。每增加一个订单，b\_id加一; foreign key包括b\_userid, b\_trainid, b\_startsid, b\_endsid, 分别与userinfo、train、station表相连。

CREATE TABLE booking (

b\_id integer NOT NULL,

b\_userid character(18),

b\_trainid character varying(5),

b\_date date NOT NULL,

b\_startsid integer NOT NULL,

b\_endsid integer NOT NULL,

b\_price numeric(5,1),

b\_status status\_type

);

1. 车次train表：包括车次的TrainID, 车次的经停站点数，车次的固有车票数（每种座位若本车次有座则为5，若无则为-1，此数在运行的整个过程中为不变的常数，查余票时在本车次的每个区间上，用这个基本常数减去记录下的当前区间已订车票数）。

CREATE TABLE train (

t\_id character varying(5) NOT NULL,

t\_numyz integer DEFAULT 5,

t\_numrz integer DEFAULT 5,

t\_numyw integer DEFAULT 5,

t\_numrw integer DEFAULT 5,

t\_stationnum smallint NOT NULL

);

1. 火车route表：每条记录为一辆列车的经停站（如下示意图所示）。

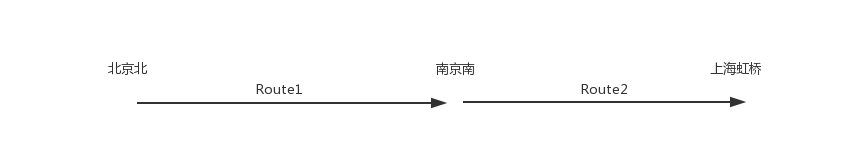


图 2 从始发站开始逐个编号

CREATE TABLE route (

r\_trainid character varying(5) NOT NULL, //该线路从属的车次编号

r\_num smallint NOT NULL, //经停站编号

r\_station integer, //经停站站点编号

r\_timedep interval, //当前站点离站时间

r\_timearr interval, //当前到站时间

r\_priceyz numeric(5,1), //以下为多种座位的票价，从始发站算起

r\_pricerz numeric(5,1),

r\_priceyws numeric(5,1),

r\_priceywz numeric(5,1),

r\_priceywx numeric(5,1),

r\_pricerws numeric(5,1),

r\_pricerwx numeric(5,1)

);

1. 站点station表：包括站点名称、所在城市、站点编号信息。直接从all-stations.txt中读出

CREATE TABLE station (

s\_id integer NOT NULL,

s\_name character varying(20),

s\_city character varying(20)

);

1. 车票ticket表：包括车票从属的车次、订单，车票的起止站点，车票日期和座位类型。

CREATE TABLE ticket (

t\_trainid character varying(5) NOT NULL,

t\_bookid integer NOT NULL,

t\_rnumdep smallint, //车票出发站

t\_rnumarr smallint, //车票到达站

t\_date date,

t\_typeticket seat\_type

);

1. 用户userinfo表：包括一切用户信息。

CREATE TABLE userinfo (

u\_id character(18) NOT NULL,

u\_realname character varying(20) NOT NULL,

u\_phoneno character(11) NOT NULL,

u\_credno character(16) NOT NULL,

u\_username character varying(20) NOT NULL,

u\_type user\_type DEFAULT 'customer'::user\_type

);

1. 范式细化

为满足所有列的原子性，我们将每车次的每两站之间记为一个区间，每个区间单独列为一份记录。如此，则不会出现违反原子性的属性。

由下表可以看出，只有route、ticket表的compoud key有可能产生部分依赖，经过检查，所有非键均需要compound key的组合唯一才能确定；因此不存在部分依赖。

又因为在我们的设计中依赖关系只会依赖于主键，不会依赖于包含非键属性的X，所以不存在非键传递依赖。

表 1 表和它们的键的情况

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | route | station | ticket | booking | user | train |
| 主键 | R\_TrainID  R\_Num | S\_ID | T\_TrainID  T\_BookID | B\_ID | U\_ID | T\_TrainID |
| 外键 | R\_StationID  R\_TrainID | / | T\_TrainID  T\_BookID | B\_UserID  B\_TrainID  B\_StationDep  B\_StationArr | U\_PhoneNo  U\_UserName | / |
| 非键 | r\_timedep  r\_timearr  r\_priceyz  r\_pricerz  r\_priceyws  r\_priceywz  r\_priceywx  r\_pricerws  r\_pricerwx | S\_Name  S\_City | T\_RNumDep  T\_RNumArr  T\_Date  T\_TypeTicket | B\_UserID  B\_TrainID  B\_Date  B\_StationDep  B\_StationArr  B\_Price  B\_Status | U\_RealName  U\_CredNo  U\_Type | T\_NumYZ  T\_NumRZ  T\_NumYW  T\_NumRW |

1. SQL查询模板
2. 需求3: 将register.html中的输入读入到register.php中，对输入数据进行处理。将数据按顺序插入userinfo表中。

<?php

$ins = <<<EOF

INSERT INTO

userinfo(u\_id, u\_realname, u\_phoneno, u\_credno, u\_username, u\_type)

VALUES ('**$U\_ID**', '**$U\_RealName**', '**$U\_PhoneNo**', '**$U\_CredNo**', '**$U\_UserName**', '**$U\_Type**');

EOF;

?>

1. 需求4：
2. $qtrain查找$TID对应的车次

$qtrain = <<<EOF

SELECT r\_trainid,r\_num,s\_name,r\_timearr,r\_timedep,r\_priceyz,r\_pricerz,r\_priceyws,r\_priceywz,r\_priceywx,r\_pricerws,r\_pricerwx

FROM

route,station

WHERE

r\_trainid = '$TID' and

r\_station = s\_id;

EOF;

1. $qticket查询当前车次、日期对应的票的初始情况

$qticket = <<<EOF

SELECT

T\_Typeticket,T\_RNumDep,Count(\*) as Number

FROM

Ticket

WHERE

T\_TrainID = '$TID' and

T\_Date = '$fdate'

GROUP BY

T\_RNumDep, T\_Typeticket

ORDER BY

T\_Typeticket,T\_RNumDep;

EOF;

然后相当于是用原有的总票数减去了当前已订的票数，得到余票。

$avlt = 5-$nticket[(string)$crow][$i];

1. 需求5：起止站点查询

<?php

$qroute = <<<EOF

set intervalstyle to iso\_8601;

SELECT

RD.R\_TrainID,

SD.S\_Name as NameL,RD.R\_TimeDep,

SA.S\_Name as NameR, RA.R\_TimeArr,

RD.r\_priceyz as yzl,RD.r\_pricerz as rzl ,RD.r\_priceyws as ywsl,RD.r\_priceywz as ywzl ,RD.r\_priceywx as ywxl,RD.r\_pricerws as rwxl ,RD.r\_pricerwx as rwxl,

RA.r\_priceyz as yzr,RA.r\_pricerz as rzr ,RA.r\_priceyws as ywsr ,RA.r\_priceywz as ywzr ,RA.r\_priceywx as ywxr,RA.r\_pricerws as rwsr ,RA.r\_pricerwx as rwxr,

to\_char( ( date '**$Date**' - extract(day from justify\_hours(RD.R\_TimeDep) )\*(interval '1 day') ),'YYYY-MM-DD') as fdate,

RD.R\_Num as numl ,RA.R\_Num as numr

FROM

route as RD, route as RA, station as SD, station as SA

WHERE

SD.S\_City = '**$CityL**' and SA.S\_City = '**$CityR**' and

SD.S\_ID = RD.R\_Station and SA.S\_ID = RA.R\_Station and

RD.R\_TrainID = RA.R\_TrainID and

RD.R\_Num < RA.R\_Num and

RD.R\_TimeDep - extract(day from justify\_hours(RD.R\_TimeDep) )\*(interval '1 day') > interval '**$Time**'

ORDER BY

(RD.R\_TimeDep - extract(day from justify\_hours(RD.R\_TimeDep) )\*(interval '1 day')) asc

LIMIT 10;

EOF;

?>

1. 需求6：预订车票：
2. 在之前程序传过来的值中记录下本次订单的TID和起止站点，然后以这些为限制条件进行SELECT操作。

$qtr = <<<EOF

set intervalstyle to iso\_8601;

SELECT

R\_Num, R\_Station, r\_priceyz, r\_pricerz, r\_priceyws, r\_priceywz, r\_priceywx, r\_pricerws, r\_pricerwx, S\_Name, R\_TimeDep, R\_TimeArr

FROM

route, station

WHERE

S\_ID = R\_Station and

R\_TrainID = '$TID' and

(R\_Num = '$T\_RNumDep' or R\_Num = '$T\_RNumArr')

ORDER BY

R\_Num;

EOF;

1. 将计算完成的数据插入到Booking表里，成为历史订单的一部分。

$qorder = <<<EOF

BEGIN;

insert into Booking(B\_UserID,B\_Date,B\_StartSID,B\_EndSID,B\_Status,B\_Price)

values('$UID','$Date','$B\_StartSID','$B\_EndSID','normal',$TotalPrice);

EOF;

for ($i=0;$i<$data['num'];$i++){

$TID = $data[$i]['TID'];

$T\_RNumDep = $data[$i]['L'];

$T\_RNumArr = $data[$i]['R'];

$T\_Date = $data[$i]['Date'];

$T\_TypeTicket = $data[$i]['SeatType'];

$qorder = $qorder. <<<EOF

insert into Ticket

values('$TID',currval('booking\_b\_id\_seq'),'$T\_RNumDep','$T\_RNumArr','$T\_Date','$T\_TypeTicket');

EOF;

}

1. 需求7：历史订单查询：

$qorder = <<<EOF

SELECT

B\_ID, B\_Date,D.S\_Name,A.S\_Name as snamel,B\_Price as snamer,B\_Status

FROM

userinfo, Booking, Station as A, Station as D

WHERE

U\_UserName = '$FName' and

B\_UserID = U\_ID and

A.S\_ID = B\_StartSID and

D.S\_ID = B\_EndSID $querydate

ORDER BY

B\_Date;

EOF;

1. 需求8：管理员界面
2. 查询总订单数，总车票价格

$qbook = <<<EOF

SELECT

count(\*), sum(B\_Price)

FROM

booking

WHERE

B\_STATUS = 'normal' or

B\_STATUS = 'expired';

EOF;

1. 查询最热的十个车次

$qticket = <<<EOF

SELECT

T\_TrainID, count(\*)

FROM

Ticket

GROUP BY

T\_TrainID

ORDER BY

count(\*) desc

LIMIT 10;

EOF;

1. 查询所有乘客信息

$quser = <<<EOF

SELECT

\*

FROM

userinfo

Order By

U\_UserName, U\_UserName;

EOF;