实验 2 报告

组员 龚仕奇，潘澳旋，刘欣宇

一、ER 图

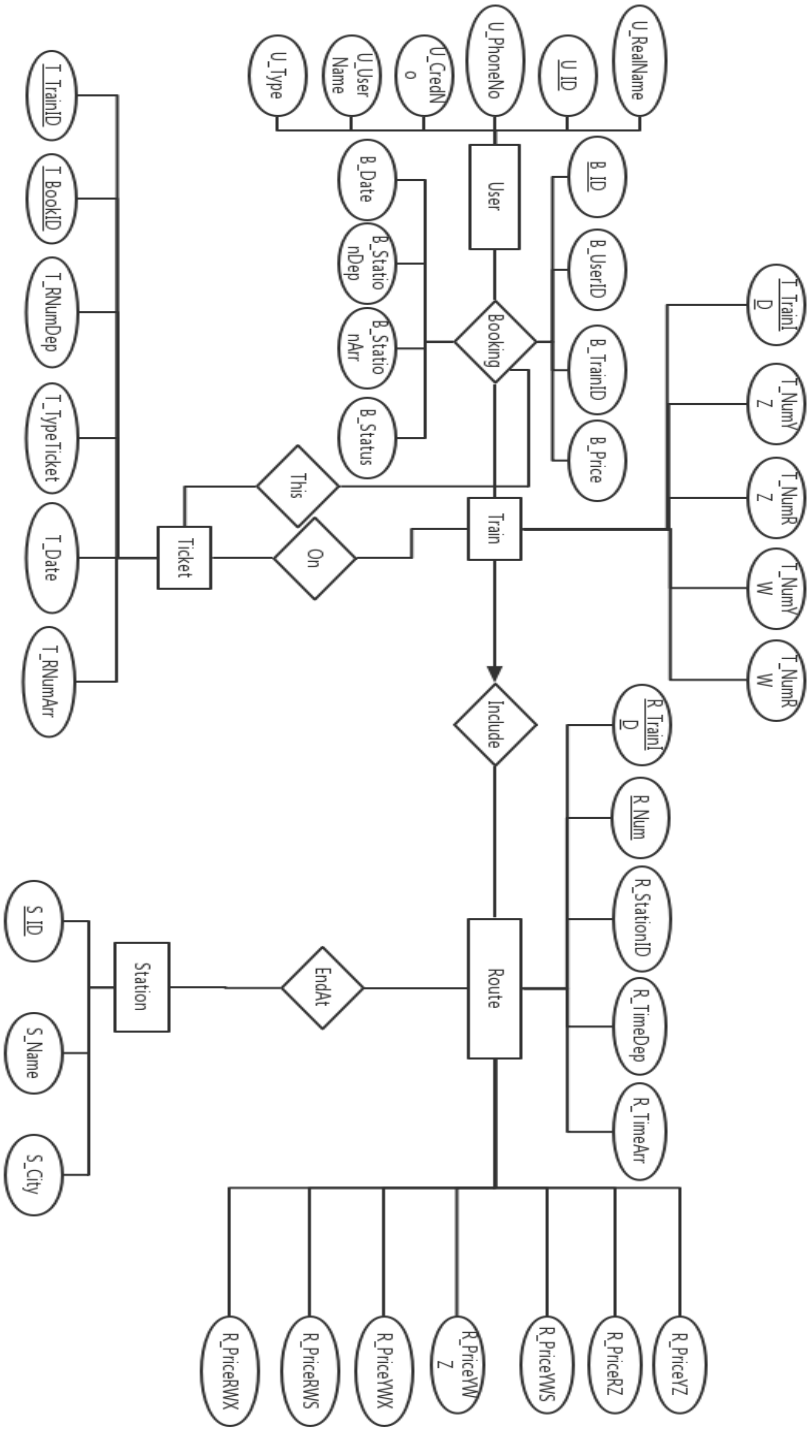

图 1 ER 图

二、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
);
```

2) 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
);
```

3) 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)
);
```

4) 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)
);
```

5) 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
);
```

6) 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
);
```

三、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

四、SQL 查询模板

1. 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;
?>
```

2. 需求 4：

1) $qtrain 查找$TID 对应的车次

```
$qtrain = <<<EOF
    SELECT
r_trainid,r_num,s_name,r_timearr,r_timedep,r_priceyz,r_pricerz,r_priceyws,r_priceywz,r_pricey
wx,r_pricerws,r_pricerwx
    FROM
        route,station
    WHERE
        r_trainid = '$TID' and
        r_station = s_id;
```

EOF;

2) $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];
```

3. 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;
?>
```

4. 需求 6：预订车票：

1) 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;
}
```

5. 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;
```

6. 需求 8：管理员界面

1) 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;
```

2) 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;
```

3) 查询所有乘客信息

```
$quser = <<<EOF
        SELECT
                *
        FROM
                userinfo
        Order By
                U_UserName, U_UserName;
EOF;
```