用户相关

- 用户：登录系统的一个账号，是购买车票的主体。
- 乘客：真实存在的一个个体，是乘坐火车的主体。

用户信息表 users

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			用户编码	
name	varchar(10)				用户昵称	
type	tinyint				用户类型	0-用户, 1-管理员
telephone	varchar(11)				联系方式	
password	varchar(20)				用户密码	
state	tinyint				用户状态	0-正常, 1-冻结, 2-注销
create_time	date				创建时间	yyyy-mm-dd

```
CREATE TABLE users (
    `id`          INT UNSIGNED AUTO_INCREMENT,
    `name`        VARCHAR(10) NOT NULL,
    `type`        TINYINT NOT NULL COMMENT '0-用户, 1-管理员',
    `telephone`   VARCHAR(11) NOT NULL,
    `password`    VARCHAR(20) NOT NULL,
    `state`       TINYINT NOT NULL COMMENT '0-正常, 1-冻结, 2-注销',
    `create_time` DATE NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE = InnoDB CHARSET = utf8;
```

乘客信息表 passengers

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			乘客编码	
name	varchar(20)				乘客姓名	
telephone	varchar(11)				联系方式	
ID_type	tinyint				证件类型	0-中华人民共和国居民身份证 1-港澳居民来往内地通行证 2-台湾居民来往大陆通行证
ID_no	varchar(30)				证件号码	
type	tinyint				乘客类型	0-成人, 1-学生, 2-儿童, 3-残疾军人

```
CREATE TABLE passengers (
    `id`         INT UNSIGNED AUTO_INCREMENT,
    `name`       VARCHAR(10) NOT NULL,
    `telephone` VARCHAR(11) NOT NULL,
    `ID_type`    TINYINT NOT NULL COMMENT '0-中华人民共和国居民身份证，1-港澳台居民 来往内地通行证，2-台湾居民来往大陆通行证',
    `ID_no`      VARCHAR(30) NOT NULL,
    `type`       TINYINT NOT NULL COMMENT '0-成人，1-学生，2-儿童，3-残疾军人',
    PRIMARY KEY (`id`)
) ENGINE = InnoDB CHARSET = utf8;
```

车站相关

省份编码表 provinces

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			省份编码	
name	varchar(10)				省份名称	

```
CREATE TABLE provinces(
    `id`   INT UNSIGNED AUTO_INCREMENT,
    `name` VARCHAR(10) NOT NULL,
    PRIMARY KEY ( `id` )
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

城市编码表 cities

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			城市编码	
province_id	int		省份编码表		所在省份编码	
name	varchar(20)				城市名称	

```
CREATE TABLE cities(
    `id`          INT UNSIGNED AUTO_INCREMENT,
    `province_id` INT UNSIGNED NOT NULL,
    `name`        VARCHAR(20) NOT NULL,
    PRIMARY KEY ( `id` ),
    FOREIGN KEY ( `province_id` ) REFERENCES provinces( `id` )
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE VIEW view_cities
AS (
    SELECT ct.id   AS id,
           pv.name AS province,
           ct.name AS city
    FROM provinces pv,
         cities ct
    WHERE pv.id = ct.province_id
    ORDER BY province
);
```

火车站编码表 stations

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			火车站编码	
city_id	int		城市编码表		所在城市编码	
station_name	varchar(20)				火车站名称	

```
CREATE TABLE stations(
    `id`      INT UNSIGNED AUTO_INCREMENT,
    `city_id` INT UNSIGNED NOT NULL,
    `name`    VARCHAR(20) NOT NULL,
    PRIMARY KEY ( `id` ),
    FOREIGN KEY ( `city_id` ) REFERENCES cities( `id` )
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE VIEW view_stations
AS(
    SELECT st.id   AS id,
           pv.name AS province,
           ct.name AS city,
           st.name AS station
    FROM provinces pv,
         cities    ct,
         stations  st
    WHERE pv.id = ct.province_id
      AND ct.id = st.city_id
    ORDER  BY pv.name, ct.name
);
```

火车相关

- 火车信息表-火车：实指一列火车实体。
- 火车运行时刻表：指该列火车在某天运行的具体行为。

火车信息表 trains

- 表中的出发时间，规定的是每次发车的固定时间，因此不存在具体日期。
- 同时可能存在次日达的情况，到达日期和出发日期不在同一天，为记录方便，表中记录的是列车的运行时长，而非到达时间。

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			火车编码	
name	varchar(10)				火车名称	eg: G1818
type	tinyint				火车类型	0-普通旅客列车 1-普通动车组列车 2-高速动车组列车 3-其他
departure_station	int		火车站编码表		起始站编码	首程起始站 返程目的站
destination_station	int		火车站编码表		目的站编码	首程目的站 返程起始站
departure_time	time				出发时间	
last_time	time				运行时长	

```
CREATE TABLE trains (
    `id`                  INT UNSIGNED AUTO_INCREMENT,
    `name`                VARCHAR(10) NOT NULL,
    `type`                TINYINT NOT NULL COMMENT '0-普通旅客列车, 1-普通动车组列车,
2-高速动车组列车, 3-其他',
    `departure_station`   INT UNSIGNED NOT NULL,
    `destination_station` INT UNSIGNED NOT NULL,
    `departure_time`      TIME NOT NULL,
    `last_time`           TIME NOT NULL,
    PRIMARY KEY ( `id`),
    FOREIGN KEY ( `departure_station` ) REFERENCES stations ( `id` ),
    FOREIGN KEY ( `destination_station` ) REFERENCES stations ( `id` )
) ENGINE = InnoDB CHARSET = utf8;
```

```
CREATE VIEW view_trains
AS (
    SELECT trains.id                                              AS id,
           trains.name                                            AS name,
           trains.type                                            AS type,
           Concat(dep.province, '/', dep.city, '/', dep.station) AS origin,
           Concat(des.province, '/', des.city, '/', des.station) AS destination,
           trains.departure_time                                  AS
departure_time,
           trains.last_time                                       AS last_time
    FROM trains,
         view_stations des,
```

```
        view_stations dep
    WHERE trains.departure_station = dep.id
      AND trains.destination_station = des.id
);
```

火车运行表 runnings

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			火车运行编码	
train_id	int		火车信息表		火车编码	
departure_date	date				发车日期	
actual_departure_time	datetime			是	实际发车时间	可为空，为空时表示未发车
actual_arrive_time	datetime			是	实际到达时间	课为空，为空时表示未到达
cancel	tinyint				是否取消	01表示

```
CREATE TABLE runnings (
    `id`                       INT UNSIGNED AUTO_INCREMENT,
    `train_id`                 INT UNSIGNED NOT NULL,
    `departure_date`           DATE NOT NULL,
    `actual_departure_datetime` DATETIME,
    `actual_arrive_datetime`    DATETIME,
    `cancel`                   TINYINT NOT NULL COMMENT '0-正常, `1-取消`',
    PRIMARY KEY ( `id` ),
    FOREIGN KEY ( `train_id` ) REFERENCES trains ( `id` )
) ENGINE = InnoDB CHARSET = utf8;
```

```
CREATE VIEW view_runnings
AS (
    SELECT runnings.id
                AS id,
           trains.id
                AS train_id,
           trains.name
                AS name,
           trains.type
                AS type,
```

```
            trains.origin
                  AS origin,
            trains.destination
                   AS destination,
            Concat(runnings.departure_date, ' ', trains.departure_time)
                  AS departure_datetime,
            Addtime(Concat(runnings.departure_date, ' ', trains.departure_time),
trains.last_time) AS arrive_datetime,
            runnings.actual_departure_datetime
                   AS actual_departure_datetime,
            runnings.actual_arrive_datetime
                  AS actual_arrive_datetime,
            runnings.cancel
                  AS cancel
    FROM    view_trains trains,
            runnings
    WHERE   trains.id = runnings.train_id
);
```

火车价位表 price

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			编码	
train_id	int		火车信息表		火车编码	
type	varchar(5)				车厢及座位类型	车厢类型直接规定该车厢内的所有座位类型
price	float				价格	

```
CREATE TABLE price (
    `id`       INT UNSIGNED AUTO_INCREMENT,
    `train_id` INT UNSIGNED NOT NULL,
    `type`     VARCHAR(5) NOT NULL,
    `price`    FLOAT NOT NULL,
    PRIMARY KEY ( `id` ),
    FOREIGN KEY ( `train_id` ) REFERENCES trains ( `id` )
) ENGINE = InnoDB CHARSET = utf8;
```

车厢座位相关

注：【车厢信息表】【座位模板】是每列火车拥有的固定的信息，无论哪次运行，都应该遵守的。座位信息表则是某列火车某一次具体运行中的每一个座位的信息。

车厢信息表 cabins

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			车厢编码	
train_id	int		火车信息表		火车编码	
type	int		火车价位表		车厢类型	
number	tinyint				车厢号	

```
CREATE TABLE cabins (
    `id`       INT UNSIGNED AUTO_INCREMENT,
    `train_id` INT UNSIGNED NOT NULL,
    `type`     INT UNSIGNED NOT NULL,
    `number`   TINYINT NOT NULL,
    PRIMARY KEY ( `id` ),
    FOREIGN KEY ( `train_id` ) REFERENCES trains ( `id` ),
    FOREIGN KEY ( `type` ) REFERENCES price ( `id` )
) ENGINE = InnoDB CHARSET = utf8;
```

```
CREATE VIEW view_cabins
AS (
    SELECT cabins.id                  AS id,
           trains.id                  AS train_id,
           cabins.number              AS cabin_number,
           price.type                 AS cabin_type,
           price.price                AS price
    FROM   view_trains trains, cabins, price
    WHERE  trains.id = cabins.train_id
       AND cabins.type = price.id
);
```

座位模板 seat_template

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			座位模板编码	
cabin_id	int		车厢信息表		车厢编码	
position	varchar(5)				座位号	eg: 16排A

```
CREATE TABLE seat_template (
    `id`       INT UNSIGNED AUTO_INCREMENT,
    `cabin_id` INT UNSIGNED NOT NULL,
    `position` VARCHAR(5) NOT NULL,
    PRIMARY KEY ( `id` ),
    FOREIGN KEY ( `cabin_id` ) REFERENCES cabins ( `id` )
) ENGINE = InnoDB CHARSET = utf8;
```

```
CREATE VIEW view_seat_template
AS (
    SELECT seat.id                    AS id,
           trains.id                  AS train_id,
           cabins.id                  AS cabin_id,
           cabins.number              AS cabin_number,
           seat.position              AS position,
           price.type                 AS seat_type,
           price.price                AS price
    FROM   view_trains trains, cabins, price, seat_template seat
    WHERE  trains.id = cabins.train_id
       AND cabins.type = price.id
       AND cabins.id = seat.cabin_id
);
```

```
DROP PROCEDURE IF EXISTS add_cabin_and_seat;

delimiter //

CREATE PROCEDURE add_cabin_and_seat(
    IN train INT,
    IN type INT,
    IN no_start INT,
    IN no_end INT,
    IN rows INT,
    IN cols INT
)
BEGIN
    DECLARE cabin INT DEFAULT 1;
    DECLARE i INT DEFAULT 1;
    DECLARE j INT DEFAULT 1;
    DECLARE k INT DEFAULT no_start;
    WHILE k <= no_end DO
        INSERT INTO cabins VALUES (null, train, type, k);
        WHILE i <= rows DO
            SET j = 1;
            WHILE j <= cols DO
                SELECT max(id) INTO cabin from cabins;
                INSERT INTO seat_template VALUES (null, cabin, Concat(i, '排',
j));
                SET j = j + 1;
            END WHILE;
            SET i = i + 1;
        END WHILE;
        SET k = k + 1;
    END WHILE;
END //
```

```
delimiter ;
```

座位信息表 seats

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			座位编码	
runnings_id	int		火车运行表		火车编码	
cabin_id	int		车厢信息表		车厢编码	
position	varchar(5)				座位号	eg: 16排A

```
CREATE TABLE seats (
    `id`         INT UNSIGNED AUTO_INCREMENT,
    `running_id` INT UNSIGNED NOT NULL,
    `cabin_id`   INT UNSIGNED NOT NULL,
    `position`   VARCHAR(5) NOT NULL,
    PRIMARY KEY ( `id` ),
    FOREIGN KEY ( `running_id` ) REFERENCES runnings ( `id` ),
    FOREIGN KEY ( `cabin_id` ) REFERENCES cabins ( `id` )
) ENGINE = InnoDB CHARSET = utf8;
```

```
CREATE VIEW view_seats
AS (
    SELECT seats.id                                          AS id,
           runnings.id                                       AS running_id,
           runnings.name                                     AS name,
           runnings.type                                     AS type,
           runnings.origin                                   AS origin,
           runnings.destination                              AS destination,
           runnings.departure_datetime                       AS departure_datetime,
           runnings.arrive_datetime                          AS arrive_datetime,
           Concat(cabins.number, '车厢 ', seats.position)    AS seat_position,
           price.type                                        AS seat_type,
           price.price                                       AS price
    FROM   view_runnings runnings,
           cabins, price, seats
    WHERE  cabins.type = price.id
       AND cabins.id = seats.cabin_id
       AND seats.running_id = runnings.id
);
```

订单相关

订单信息表 orders

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			订单编码	
user_id	int		用户信息表		购票人（用户）编号	
create_time	datetime				订单创建时间	
paid	tinyint				是否已经付款	1-成功付款
cancel	tinyint				是否取消订单	1-取消订单

```
CREATE TABLE orders (
    `id`          INT UNSIGNED AUTO_INCREMENT,
    `user_id`     INT UNSIGNED NOT NULL,
    `create_time` DATETIME NOT NULL,
    `paid`        TINYINT NOT NULL,
    `cancel`      TINYINT NOT NULL,
    PRIMARY KEY ( `id` ),
    FOREIGN KEY ( `user_id` ) REFERENCES users ( `id` )
) ENGINE = InnoDB CHARSET = utf8;
```

```
CREATE VIEW view_orders
AS (
    SELECT orders.id AS id,
           users.id AS user_id,
           users.name AS user_name,
           orders.create_time AS create_time,
           orders.paid AS paid,
           orders.cancel AS cancel
    FROM users, orders
    WHERE users.id = orders.user_id
);
```

订单详细 details

字段名称	字段类型	是否主键	是否外键	是否为空	字段含义	备注
id	int	是			订单详细编码	
order_id	int	是	订单信息表		订单编码	
passenger_id	int		乘客信息表		乘客编码	
seat_id	int		座位信息表		座位编码	
change	int		订单详细	是	订单详细编码	为空时表示没有改签 不为空时值表示改签后订单详细编码
refund	tinyint				是否退票	1-退票

```
CREATE TABLE details (
    `id`           INT UNSIGNED AUTO_INCREMENT,
    `order_id`     INT UNSIGNED NOT NULL,
    `passenger_id` INT UNSIGNED NOT NULL,
    `seat_id`      INT UNSIGNED NOT NULL,
    `change`       INT UNSIGNED,
    `refund`       TINYINT NOT NULL,
    PRIMARY KEY ( `id` ),
    FOREIGN KEY ( `order_id` ) REFERENCES orders ( `id` ),
    FOREIGN KEY ( `passenger_id` ) REFERENCES passengers ( `id` ),
    FOREIGN KEY ( `seat_id` ) REFERENCES seats ( `id` ),
    FOREIGN KEY ( `change` ) REFERENCES details ( `id` )
) ENGINE = InnoDB CHARSET = utf8;
```

```
CREATE VIEW view_details
AS (
    SELECT details.id AS id,
           orders.id AS order_id,
           Concat(details.passenger_id, '/', passengers.name) AS passenger,
           seats.running_id AS running_id,
           seats.name AS train,
           seats.origin AS origin,
           seats.destination AS destination,
           seats.departure_datetime AS departure_time,
           seats.arrive_datetime AS arrive_time,
           seats.seat_position AS seat_position,
           seats.seat_type AS seat_type,
           seats.price AS price,
           orders.cancel AS order_cancel,
           details.change AS detail_change,
           details.refund AS detail_refund
    FROM  orders,
          details,
          passengers,
```

```
            view_seats seats
    WHERE orders.id = details.order_id
      AND details.passenger_id = passengers.id
      AND details.seat_id = seats.id
);
```

```
CREATE VIEW view_valid_details
AS (
    SELECT details.*
    FROM   orders, details
    WHERE  orders.id = details.order_id
       AND orders.cancel = 0
       AND details.change IS NULL
       AND details.refund = 0
);
```

操作流程

添加用户和乘客

- `users` 用户信息表添加用户 `(null, '用户名', 用户类型, '联系方式', '用户密码', 用户状态, 创建时间)`

```
INSERT INTO users VALUES (null, 'test01', 0, '12345678911', 'admin', 0, NOW());
```

- `passengers` 乘客信息表添加乘客 `(null, '姓名', '联系方式', 证件类型, '证件号码', 乘客类型)`

```
INSERT INTO passengers VALUES (NULL, '苏桐渤', '12345678911', 0,
'1234***********789', '1');
```

添加火车

- `provinces` 省份编码表添加对应省份 `(null, '省份')`

```
INSERT INTO provinces VALUES(NULL, '浙江省');
```

- `cities` 城市编码表添加对应城市 `(null, 省份编码, '城市')`

```
INSERT INTO cities VALUES(NULL, 1, '杭州市');
```

- `stations` 火车站编码表添加车站 `(null, 城市编码, '车站')`

```
INSERT INTO stations VALUES(null, 1, '杭州东站');
```

- `trains` 火车信息表添加火车 `(null, '火车', 火车类型编码, 起始站编码, 目的站编码, '发车时间', '运行时长')`

```
INSERT INTO trains VALUES (null, 'D3111', 1, 1, 6, '07:30:00', '00:19:00');
```

添加固定车厢和座位

- `price` 火车价位表添加对应的座位类型及价格 `(null, 火车编码, '座位类型', 价格)`

```
INSERT INTO price VALUES (null, 10, '软卧', 112.5);
INSERT INTO price VALUES (null, 10, '硬卧', 74.5);
INSERT INTO price VALUES (null, 10, '硬座', 28.5);
```

- `cabins` `seat_template` 添加固定的车厢和座位模板
- 存储过程 `add_cabin_and_seat(火车编码, 车厢/座位类型, 起始车厢, 结束车厢, 行数, 列数)`

```
CALL add_cabin_and_seat(10, 26, 1, 1, 1, 1);
CALL add_cabin_and_seat(10, 27, 2, 3, 2, 2);
CALL add_cabin_and_seat(10, 28, 4, 5, 3, 3);
```

添加运行班次和相应座位

- `tunnings` 火车运行表添加火车的某次运行 `(null, 火车编码, '发车日期', null, null, 0)`

```
INSERT runnings VALUES (NULL, 1, '2021-12-22', NULL, NULL, 0);
INSERT runnings VALUES (NULL, 2, '2021-12-22', NULL, NULL, 0);
INSERT runnings VALUES (NULL, 3, '2021-12-22', NULL, NULL, 0);
INSERT runnings VALUES (NULL, 4, '2021-12-22', NULL, NULL, 0);
INSERT runnings VALUES (NULL, 5, '2021-12-22', NULL, NULL, 0);
INSERT runnings VALUES (NULL, 6, '2021-12-22', NULL, NULL, 0);
INSERT runnings VALUES (NULL, 7, '2021-12-22', NULL, NULL, 0);
INSERT runnings VALUES (NULL, 8, '2021-12-22', NULL, NULL, 0);
INSERT runnings VALUES (NULL, 9, '2021-12-22', NULL, NULL, 0);
INSERT runnings VALUES (NULL, 10, '2021-12-22', NULL, NULL, 0);
```

- `seats` 座位信息表添加一次运行的所有座位，选择对应的运行编码

```
INSERT INTO seats (
    SELECT null, runnings.id, seat.cabin_id, seat.position
    FROM   runnings, cabins, seat_template seat
    WHERE  cabins.id = seat.cabin_id
       AND runnings.train_id = cabins.train_id
       AND runnings.id = 3
);
```

查询余票

- `view_runnings` 查询当日运行的相关火车

```
SELECT *
FROM view_runnings
WHERE departure_datetime LIKE '%12-21%'
  AND origin LIKE '%杭州%'
  AND destination LIKE '%潮汕%';
```

- `view_seats` 查询车次的余票情况

```sql
SELECT COUNT(*)
FROM view_seats
WHERE running_id = 6
  AND seat_type = '二等座'
  AND id NOT IN (SELECT seat_id FROM view_valid_detail);
```

添加订单

- `orders` 订单信息表添加新的订单 `(null, 用户编码, 时间, 0, 0)`

```sql
INSERT INTO orders VALUES (null, 1, NOW(), 0, 0);
```

- `details` 订单向西添加相应车票 `(null, 订单编码, 乘客编码, 座位编码, null, 0)`

```sql
INSERT INTO details VALUES (null, 1, 1, 69, null, 0);
```