

1 Railway Reservation System - (Redesigning IRCTC database)

**Train (trainNumber, name, source,
destination, start_time, reach_time,
traveltime, distance, class, days, type)**

**Ticket (PNRNo, Transactionid,
from_station, to_station,
date_of_journey,
class,date_of_booking,
total_ticket_fare, train number)**

**Passenger (PNR No, Serial no, Name,
Age, Reservation_status)**

**Train_Route(Train_No, route_no,
station_code, name, arrival_time,
depart_time, distance, day)**

**Train_Ticket_fare(Train_No, class,
base_fare, reservation_charge,
superfast_charge, other_charge,
tatkal_charge, service_tax)**

**Create all the tables specified above.
Make underlined columns as primary
key.(use number, number(m,n),
varchar(n), date, time, timestamp
data types appropriately)**

**Insert atleast 5 rows to each table.
(Check www.irctc.co.in website for
actual data)**

- 1. Use Interactive insertion for
inserting rows to the table.**
- 2. Use ADT (varray) for class and days
column in Train table.**

solution:

```
mysql> 1. create table Train(trainno INT(6)  
PRIMARY KEY,name VARCHAR(20),source  
VARCHAR(20),destination  
VARCHAR(20),start_time DATETIME,reach_time  
DATETIME,traveltime TIME,distance FLOAT(6,2),  
class VARCHAR(10),days INT(2),type VARCHAR(5));
```

```
mysql>2. create table Ticket (PNRNO INT(10)  
PRIMARY KEY, transactionid INT(10), from_station  
VARCHAR(20), to_station VARCHAR(20),  
date_of_journey DATETIME, class VARCHAR(10),  
date_of_booking DATETIME, total_ticket_fare  
INT(5), trainno INT(6));
```

```
mysql>3. create table Passenger (PNRNo INT(10) ,  
Serialno INT(10), Name VARCHAR(20), Age INT(3),  
Reservation_status VARCHAR(10) PRIMARY  
KEY(PNRNO,Serialno));
```

```
mysql>4. create table Train_Route(trainno INT(6),  
route_no INT(6), station_code VARCHAR(5), name  
VARCHAR(20), arrival_time TIME, depart_time  
TIME, distance FLOAT(6,2), day INT(2) PRIMARY  
KEY(trainno,route_no));
```

```
mysql>5. create table Train_Ticket_fare(trainno  
INT(6), class VARCHAR(10), base_fare INT(4),  
reservation_charge INT(4), superfast_charge  
INT(4), other_charge INT(4), tatkal_charge INT(4),  
service_tax INT(4), primary key(trainno,class));
```

Write simple DDL/DML Queries to

1. Remove all the rows from Passenger
 2. Change the name of the Passenger
 3. List all train details.
 4. List all passenger details.
 5. Give a list of trains in ascending order
 6. List the senior citizen passengers details
 7. List the station names where code is
 8. List the trains details within a range
 9. Change the super fast charge value to null.
 10. List the passenger names whose ticket
 11. List the base_fare of all AC coaches
- Find the ticket details where transaction
- 1) Use Interactive updation for updating PNR NO.
 - 2) Find the train names that are from South do not have the source or destination
 - 3) Find the train details that are on Thursday (created).

- 1 Delete * from passenger;
- 2 RENAME table passenger to passenger_Details;
- 3 Select * from train;
- 4 Select * from passenger;
- 5 Select * from train order by trainno;
- 6 Select * from passenger where age>=45;

- 7 select name from train_route where station_code LIKE 'M%';
- 8 Select * from train where trainno between 1235 and 1237;
- 9 Update train_ticket_fare set super_fast_charge=0 where super_fast_charge is NULL;
- 10 Select name from passenger where reservation_status='not conf';

3)

```
mysql> alter table ticket add foreign key(trainno) references train(trainno);
```

Query OK, 1 row affected (1.23 sec)

Records: 1 Duplicates: 0 Warnings: 0

- 1)alter table train add CHECK(trainno>=10001
and trainno<=99999)
- 2)alter table train_route modify name
varchar(30) not null;
- 3)alter table train_route modify column
arrival_time timestamp;
alter table train_route modify column
depart_time date;
- 4)alter table train add constraint chk_val
CHECK(class in ('1A','2A','3A','SL','C'));
- 5)alter table train modify distance float(6,2)
NOT NULL;

4)

1) Find the passengers whose date of journey is
one month from today.

```
select pnrno,name,date_of_journey  
from ticket t,passenger p where t.pnrno=p.pnrno  
and
```

MONTH(date_of_journey)=MOD(MONTH(CURDATE()),12)+1 from passenger p,ticket t;

2) Print the train names in upper case.

select upper(name) from train;

3) Print the passenger names with left padding character.

select pnrno,serialno,LPAD(name,20,'*') as name from passenger;

4) Print the station codes replacing K with M.

select
trainno,name,replace(station_code,'H','h') as station_code from train_route;

5) Display the fare details of all trains, if any value is ZERO, print as NULL value.

SELECT NULLIF(base_fare, 0) AS base_fare
FROM train_ticket_fare;

6) Display the pnrno and transaction id, if transaction id is null, print 'not generated'

SELECT pnrno, IF(transactionid IS NULL,'not generated') AS "transactionid" from ticket.

7) Print the date_of_journey in the format '27th November 2010'

```
SELECT  
pnrno,DATE_FORMAT(date_of_journey,'%D %M  
%Y') as date_of_journey from ticket;
```

8) Find the maximum fare (total fare).

```
select trainno,class,( base_fare +  
reservation_charge + superfast_charge +  
other_charge + tatkal_charge + service_tax ) as  
maxfare from train_ticket_fare;
```

9) Find the average age of passengers in one ticket.

```
select avg(Age) as age from passenger;
```

10) Find the maximum length of station name available in the database.

```
SELECT max(char_length(name)) from  
train_route;
```

11) Print the fare amount of the passengers as rounded value.

```
SELECT trainno, class,  
(base_fare+reservation_charge  
+superfast_charge+other_charge+tatkal_charge+  
service_tax) as fare_amount from  
train_ticket_fare;
```

12) Add the column halt time to train route.

```
alter table train_route add halt_time time;
```

13) Update values to it from arrival time and depart time.

```
update train_route set halt_time=depart_time-  
arrival_time;
```

5.

Write Queries to.

Use SET Operators

1. Find the train integers for which reservation have not yet been made.

```
select trainno from train  
minus  
select trainno from ticket;
```

2. Find the train names that donot have a first AC class coach.

```
select name from train  
EXCEPT
```

```
select name from train where class!='1AC';
```

3. Print all the PNR nos available in the database.

```
select pnrno from ticket  
union  
select pnrno from passenger;
```

4. Find passenger names who have booked to 'Pune'.

```
select p.name from passenger p,ticket t where  
t.to_station='pune' and p.pnrno=t.pnrno;
```

Use Nested Query(in Operators)

1. Find the train names that stop in 'Warangal'.

```
select t.name from train t,train_route r where  
r.name='warangal' and t.trainno=r.trainno;
```

2. Find the train names that are superfast and the service tax is zero.

```
select t.name from train t,train_ticket_fare tf
where tf.service_tax=0 and t.type='superfast';
```

3. Find the Passenger name who have booked for the train that starts from 'Secunderabad'.

```
select p.name from passenger p,ticket t,train tr
where tr.source='secunderabad' and
p.pnrno=t.pnrno and t.trainno=tr.trainno;
```

4. Find the trains names that have all the AC coaches and the base fare is less than 3000 for each case.

```
select t.name from train t,train_ticket_fare tf
where tf.base_fare<3000 and t.trainno=tf.trainno;
```