

Computation of a possible route of travel using Interclass booking in the railways

Apoorv Aditya¹, Namrata Chatterjee² and Kumar Devdutta³

ABSTRACT

Introduction of an algorithm and database design that can calculate a possible itinerary route of travel in railways using interclass booking with the help of an intermediate junction that will act as a journey breaking point for one type of class of travel such as 3AC, 2AC, Sleeper, etc. into another similar class of travel where tickets are available, so that travelling is made possible instead of no travelling when the ticket in any class from our desired source to destination is unavailable

Index Terms: algorithm, irctc, database, data mining, data structure, railway, smart computing, ticket booking

1. INTRODUCTION

Today Indian Railways has established itself as one of the core means of transportation for a common man. A common man's chariot for travelling, the railways provide the people a cheap and reliable means of transport from one station to another. If a person is travelling with the help of Indian Railways, he needs to buy a ticket, otherwise he may be fined or jailed or both. In order to save ourselves from this harsh punishment, the Indian Railways tickets are a faster sellout than most of the concert tickets or game tickets. People wait in queues for hours in the railway ticket counters just to buy one ticket. Most people fail to reach their destination because they fail to get the tickets. What they fail to realize is that they can still make it to their destination if they could find out an optimized route.

We referred to Indian Railways, Significance and problems of Indian Railways by Smriti Chand to look deeper into the problem and are now trying to design an algorithm and a database structure that will help in the efficient determination of possibility of booking a ticket using the concept of interclass booking, where a person can book his tickets in one class or category from the source to an intermediate junction and then, can book tickets from the intermediate junction to his destination station in another class of travel. Classes of travel include 3AC, 2AC, 1AC, SL, CC, 2S. The outcome includes possible options of travel for the user to choose from.

Example: If a user wants to travel from Tata to Pune, he has one train as an option which is Azad Hind Express and he is not getting a confirm ticket, it may be possible that he will get a confirm seat in Sleeper class from Tata to Nagpur and a 3AC confirm ticket from Nagpur to Pune. Our algorithm will show him these types of options where he can break the journey in the same train in different classes.

2. ASSUMPTIONS

We have assumed the following points before designing the database and the algorithm:-

- The trains will always be available and will be on time reaching every station.
- There are no reservations of any category such as Handicapped quota, etc.
- The number of tickets in a particular train from a source to destination on any day remains constant.

3. DATABASE DESIGN

Following are the tables which are in the database storing various relevant information. The information stored is used at various stages in the algorithm. Most of the stored data are taken from trusted sources like Rail bandhu, the onboard magazine of Indian Railways, Indian Railways train schedule and Report on Indian Railways by Railway Board, India. Collectively the tables contain train numbers, train names, sources, destinations, departure days and number of seats in each class between every source and destination, necessary for the computation purpose

```
mysql> select *from train_id;
```

id	train_no	train_name
1	12073	hwh bbs janshatabdi exp
2	22343	patna ernakulam sf exp
3	18477	kalinga utkal exp

Figure 1: Where information about the train number and train name is stored. A unique id is given to all trains. The id is used further in the algorithm to validate the train name and joining various other tables.

```
mysql> select *from train_dept_1;
```

id	source	destn	dept_days
1	hwh	bbs	MTuWedThuFSat
1	hwh	kgp	MTuWedThuFSat
1	hwh	bls	MTuWedThuFSat
1	kgp	bls	MTuWedThuFSat
1	kgp	bbs	MTuWedThuFSat
1	bls	bbs	MTuWedThuFSat

Figure 2: This is a sample table , where all the possible combinations of source and destination covered by the train having id 1 is stored. The departure days of the train are also stored, which are used in checking the availability of the train on a specific date and between a specific pair source and destination. Here id is the foreign key referring the id(primary key) in the table train_id. To fetch the dept_days the table train_id is joined with this table keeping id as the key.

```
mysql> select *from train_dept_2;
```

id	source	destn	dept_days
2	ers	pnbe	mtue
2	ers	mas	mtue
2	ers	bbs	mtue
2	ers	bza	mtue
2	mas	pnbe	tuew
2	mas	bza	tuew
2	mas	bbs	tuew
2	bza	bbs	tuew
2	bza	pnbe	tuew
2	bbs	pnbe	wthu

Figure 3: This is a table showing all the possible combinations of source and destination covered by the train having id 2. The table also contains the departure days. This table has the same functionality as that of train_dept_1.

```
mysql> select *from train_dept_3;
+-----+-----+-----+-----+
| id    | source | destn  | dept_days |
+-----+-----+-----+-----+
| 3     | puri   | hw     | mtuewthufsatsun |
| 3     | puri   | bbs    | mtuewthufsatsun |
| 3     | puri   | tata   | mtuewthufsatsun |
| 3     | puri   | rou    | mtuewthufsatsun |
+-----+-----+-----+-----+
```

Figure 4: This is a table showing all the possible combinations of source and destination covered by the train having id 3. The table also contains the departure days. This table has the same functionality as that of train_dept_1.

```
mysql> select *from t_ticket1_train1
+-----+-----+-----+-----+
| id    | source | destn  | s_no |
+-----+-----+-----+-----+
| 1     | hwh    | bbs    | 1    |
| 1     | hwh    | kgp    | 2    |
| 1     | hwh    | bls    | 3    |
| 1     | kgp    | bls    | 4    |
| 1     | kgp    | bbs    | 5    |
| 1     | bls    | bbs    | 6    |
+-----+-----+-----+-----+
```

Figure 5: Where information about the train source and its various destinations are assigned unique serial numbers. This serial number is used in joining this table with the table t_ticket2_train1 and hence used in fetching the number of seats available in all the classes present in the train. This is a sample table containing data only of train having id 1. Similarly there are separate tables for all the trains. Here, s-no is the primary key.

```
mysql> select *from t_ticket2_train1;
+-----+-----+-----+-----+-----+-----+
| s_no | third_ac | sec_ac | first_ac | cc  | sl |
+-----+-----+-----+-----+-----+-----+
| 1    | NULL    | NULL   | NULL    | 360 | NULL |
| 2    | NULL    | NULL   | NULL    | 72  | NULL |
| 3    | NULL    | NULL   | NULL    | 72  | NULL |
| 4    | NULL    | NULL   | NULL    | 72  | NULL |
| 5    | NULL    | NULL   | NULL    | 72  | NULL |
| 6    | NULL    | NULL   | NULL    | 72  | NULL |
+-----+-----+-----+-----+-----+-----+
```

Figure 6: Where information about the tickets for various classes according to serial numbers are located as a master table. Here s_no is the foreign key referring the s_no in t_ticket1_train1. Copies of this master table are made for each day so as to keep updating the number of seats. We keep the master table only for making copies, this table is immutable so as to avoid any kind of inconsistency in the data.

```
mysql> select *from t_ticket2_train1_20042016;
+-----+-----+-----+-----+-----+-----+
| s_no | third_ac | sec_ac | first_ac | cc | sl |
+-----+-----+-----+-----+-----+-----+
| 1 | NULL | NULL | NULL | 360 | NULL |
| 2 | NULL | NULL | NULL | 72 | NULL |
| 3 | NULL | NULL | NULL | 72 | NULL |
| 4 | NULL | NULL | NULL | 72 | NULL |
| 5 | NULL | NULL | NULL | 72 | NULL |
| 6 | NULL | NULL | NULL | 72 | NULL |
```

Figure 7: Where information about the tickets available is gained on a particular date mentioned in the table name. This is a copy of the master table t_ticket2_train1.

```
mysql> select *from t_ticket2_train2_20042016;
+-----+-----+-----+-----+-----+-----+
| s_no | third_ac | sec_ac | first_ac | cc | sl |
+-----+-----+-----+-----+-----+-----+
| 1 | 64 | 23 | NULL | NULL | 434 |
| 2 | 7 | 3 | NULL | NULL | 23 |
| 3 | 7 | 2 | NULL | NULL | 24 |
| 4 | 7 | 3 | NULL | NULL | 23 |
| 5 | 7 | 2 | NULL | NULL | 24 |
| 6 | 7 | 3 | NULL | NULL | 23 |
| 7 | 7 | 2 | NULL | NULL | 24 |
```

Figure 8: Where information about the ticket availability for various serial numbers are gained for another day of booking of the train having id 2.

```
mysql> desc user_authentication;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default |
+-----+-----+-----+-----+-----+
| user_id | varchar(20) | NO | PRI | |
| email | varchar(50) | NO | PRI | |
| password | varchar(50) | YES | | NULL |
```

Figure 9: This table is used for storing the user_id, email and the password of the user obtained at the time of registration. This is further used in validating the user. The user_id and email is the composite primary key.

```
mysql> desc user_info;
```

Field	Type	Null	Key	Default
user_id	varchar(20)	YES	MUL	NULL
email	varchar(50)	YES		NULL
name	varchar(20)	YES		NULL
phone	int(10)	YES		NULL

Figure 10: This table stores various other information of the registered users like name and password. Here user_id is the foreign key referring the composite primary key in table user_authentication. This ensures that information of only the registered users are stored avoiding data inconsistency.

```
mysql> desc user_pnr;
```

Field	Type	Null	Key	Default
user_id	varchar(20)	YES	MUL	NULL
email	varchar(50)	YES		NULL
pnr	int(10)	NO	PRI	NULL
src	varchar(10)	YES		NULL
dest	varchar(10)	YES		NULL
train_no	int(10)	YES		NULL

Figure 11: This table is updated as soon as the ticket is booked. This table keeps a record of the tickets booked. Here pnr is the primary key so as to permit only unique values of the pnr and user_id is the foreign key referring the composite primary key in table user_authentication.

```
mysql> desc pnr_info;
```

Field	Type	Null	Key	Default
pnr	int(10)	YES	MUL	NULL
date_of_booking	date	YES		NULL
date_of_journey	date	YES		NULL
no_of_seats	int(6)	YES		NULL
seat_no	varchar(20)	YES		NULL

Figure 12: This table contains all the information regarding a particular ticket, having a unique pnr. Here pnr is the foreign key referring the pnr in the table user_pnr.

4. ALGORITHM

Make the user register herself/himself. Ask for user_id, password, email_id, name and phone number. Store these information in the table user_authentication and user_info (ref. fig 9 and fig 10). At the time of login, ask for user_id or email_id and password, validate the user using user_authentication table and allow login, else redirect to register. If user is valid go to STEP A.

STEP A: user input: src, destn, preferred train name, date of travel and preferred class

step A: if preferred train name is entered, select id from table train_id(ref. fig 1).

After fetching the id from table train_id fetch dept_days from table train_dept_<id> corresponding to the given travel date(ref. fig 2). If train not available on that date, ask for any other travel date else if the train is available on that day move to next step B.

STEP B: select s_no from t_ticket1_train<id> where source and destination matches with the one entered by the user(ref. fig 5). Using s_no fetched, fetch the number of seats present in the preferred class from table t_ticket2_train<id>_<date:ddmmyyyy> (ref. fig 7).

If the number of seats is not equal to zero, give an option to the user to book the ticket. If the user books the ticket then store various information like seat_no, coach_no, journey date, date of booking, train_no and pnr in the tables user_pnr and pnr_info (ref. fig 11 and fig 12). Then update the number of seats in train_ticket2_train<1><date:ddmmyyyy>.

If the number of seats in the preferred class equal to zero, go to step ICB.

STEP ICB:

Select all the destinations from t_ticket1_train<id> where the source matches with the one entered by the user (ref. fig 5).

Store the fetched destinations as the intermediate stations.

Using table t_ticket1_train<id>, check whether the destination given by the user can be reached from the stored intermediate stations.

If yes, fetch the Number of seats available in all the classes from the source to intermediate and from the intermediate to destination separately using t_ticket2_train<id> and store in a n-d array.

Display the array to the user as list of options.

If the user selects an option, book the ticket and update the number of seats in t_ticket2_train<id><date:ddmmyyyy> and store the ticket related information in user_pnr and pnr_info.

If user does not select any option, then EXIT.

5. OUTCOME OF THE SYSTEM

After implementing this algorithm and the database structure, what we get as an output, is a list of possible choices for the user to choose and book a ticket in spite of unavailability of seats in the preferred class. For example, if a user wants to book a ticket in hwhbbsjanshatabdi express in chair car from hwh to bbs and there is no seat available for the date entered by the user, then our algorithm will compute the following choices for the user:

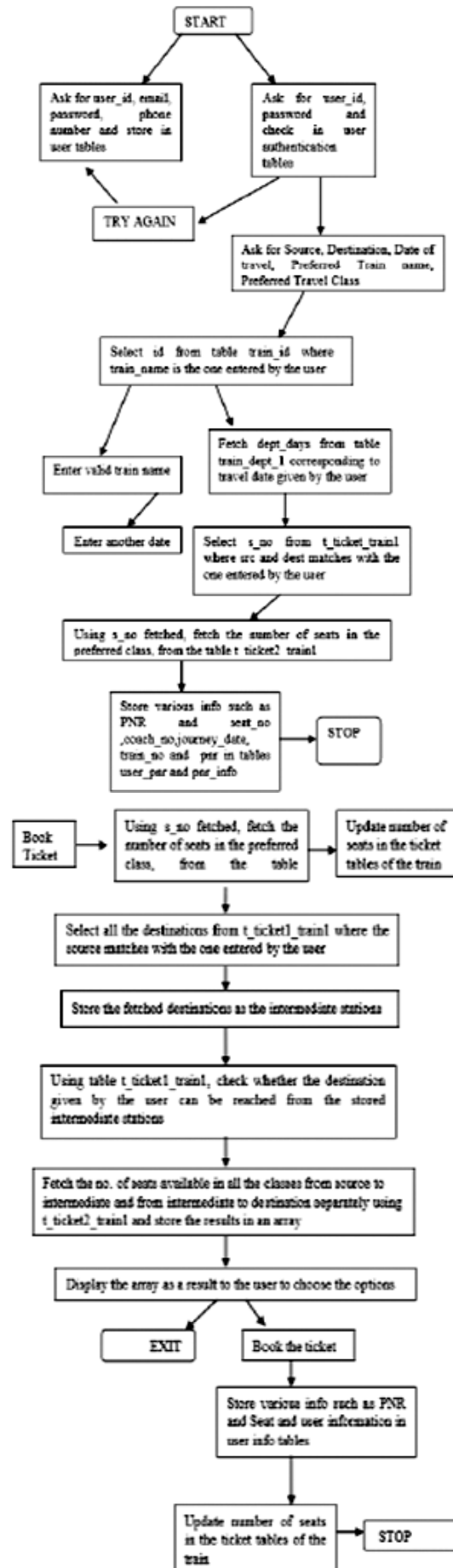
1st option

- hwh to kgp number of seats available is 6
- kgp to bbs number of seats available is 10

2nd option

- hwh to bls number of seats available is 3
- bls to bbs number of seats available is 5

6. ALGORITHM FLOWCHART



Assumptions:

- Flowchart is drawn considering train_id=1.
- For seat availability the master table is referred.

7. CONCLUSION

In this paper, we have used a database design and an efficient algorithm that effectively allows a user to find an itinerary route to travel from a source to a destination with the help of interclass booking and we have developed our algorithm and database design using some aforementioned assumptions. We have tried to implement something which has not yet been implemented by the Indian Railways. To comprehend the management of Indian Railways more clearly we referred to the Journal of Rail Transport Planning & Management by I. Hansen and Report on Indian Railways by National Government Publication.

REFERENCES

- [1] I. Hansen: Journal of Rail Transport Planning & Management, Affiliated with the International Association of Railway Operations Research.
- [2] Smriti Chand: Indian Railway, Significance and problems of Indian Railways URL- <http://www.yourarticlelibrary.com/railways/indian-railway-significance-and-problems-of-indian-railways/14136/>
- [3] Railway Board, India: Report on Indian Railways, National Government Publication, WorldCat database, includes information on the state railways from their beginning.
- [4] Ministry of Railways, Research Designs and Standard Organization, India: Indian Railway Technical Bulletin, R.D.S.O publisher, Lucknow.
- [5] Indian Railways: Train Schedule URL- www.indianrail.gov.in/train_Schedule.htm
- [6] Rail Bandhu: The Onboard Magazine of Indian Railways. URL- railbandhu.in

