Project: Tour on Wheels

Sanjali Agarwal (2018CS50420), Dipen Kumar (2018CS50098), Manoj Kumar(2018CS50411)

April 4th, 2021

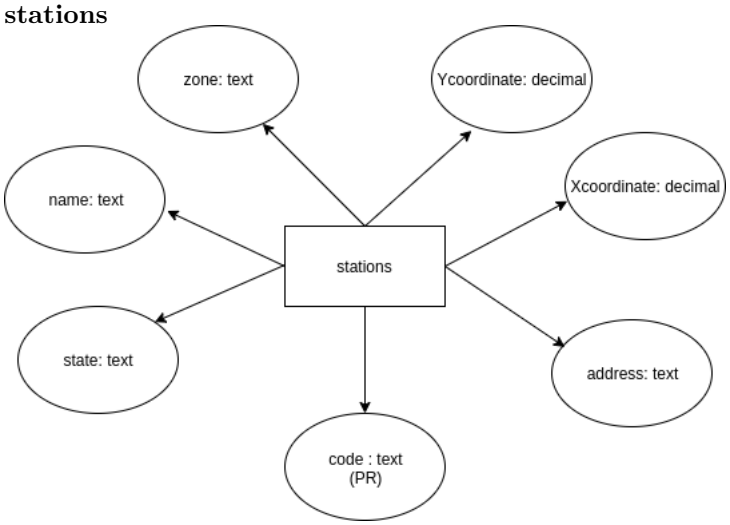
1 Section 1

Tour on Wheels will use the dataset to answer user query on trains details, available trains between stations. A User can do the following:

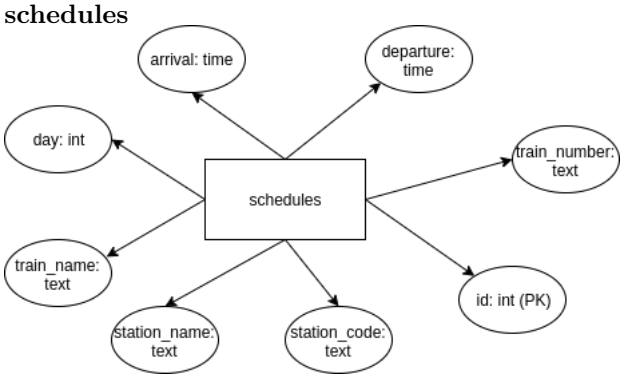
1. Users can search the trains between two stations by providing source and destination name (or choose from dropdown list) and date of journey.
2. Users can get information about the availability of seats in different classes of the train.
3. Users can book train tickets according to their preferences by providing their personal details.
4. A printed ticket will be given to the user after booking a train ticket.
5. Each booking is assigned a PNR number.
6. Users can check the status of their booking using their PNR number.
7. Users can cancel their booking using their PNR number.

Entity sets

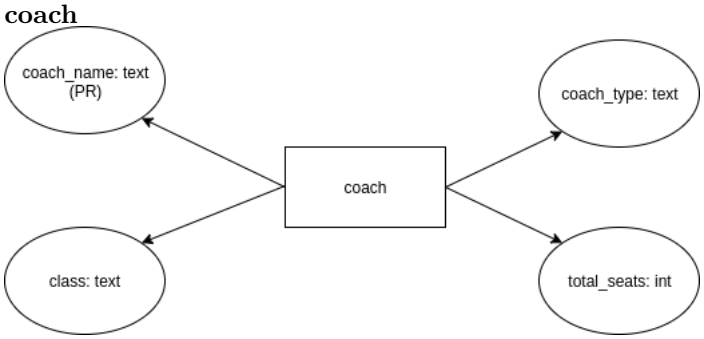
1. **stations**



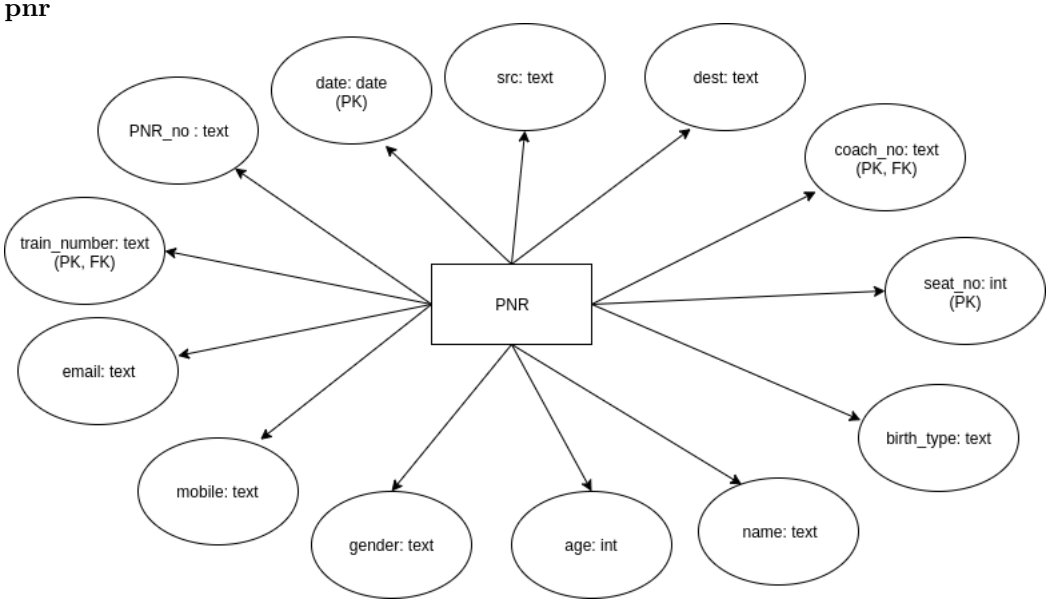
2. **schedules**



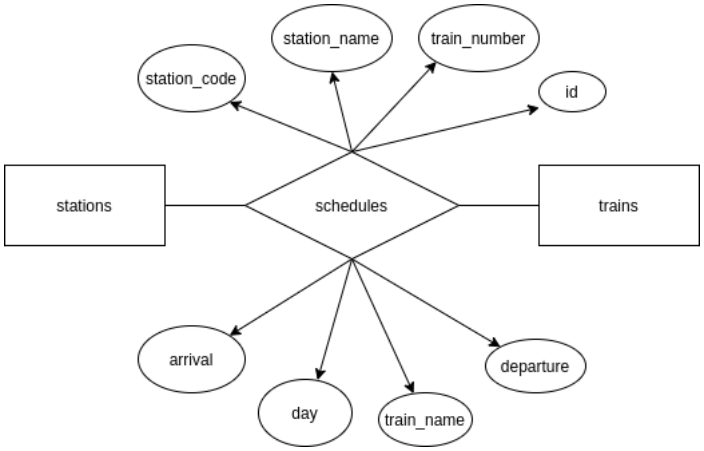
3. **coach**



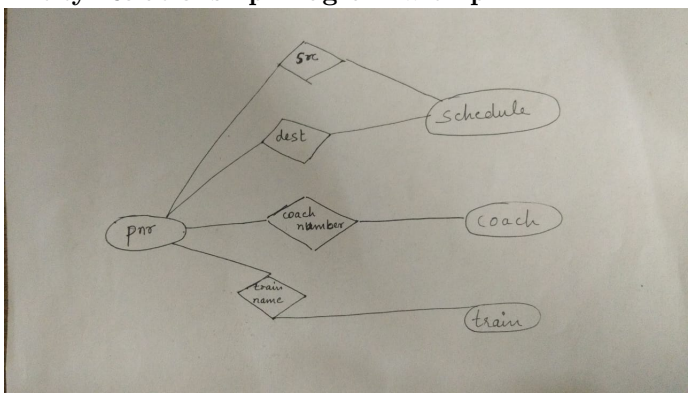
4. **pnr**



5. **Entity-Relationship Diagram between stations and trains**



6. **Entity-Relationship Diagram with pnr**



2 Section 2

1. Source of data: https://www.kaggle.com/sripaadsrinivasan/indian-railways-dataset?select=trains.json
2. data downloaded from kaggle in json format. json data is converted to csv data using python scripts and used.
3. Some additional tables have been introduced:

   (a) coach: information about all coaches in a train.

   (b) pnr: to store the tickets and entertain PNR enquiry.

   (c) view total_seats_available: information of total available seats in a particular coach of a particular train.

4. Cleanup Steps:

   (a) json files are converted to readable csv format and removed irreverent information.

   (b) Data entries for which value was 'None', they were converted to Null.

   (c) In schedules table, entries, in which both arrival and departure time is 'None' or null, are erased from database.

(d) new table created (coach) and inserted values in it.

(e) To store the tickes, New table (pnr) created.

5. **Statistics:**

table	no. of tuples	Time to load	size (raw dataset)	size (after cleanup)
stations	8990	78.909 ms	509.4KB	1.9MB
schedules	394391	14493.553 ms	82.0 MB	29.9 MB
trains	5208	760.239 ms	14.8MB	621KB

Table 1: Statistics of database

3 Section 3

User's view of the system:

1. First and introductory page of the web application is index.html, It opens up with a form to fill source, date and Destination of the journey. User can either type these inputs, or select from the dropdown menu.
2. If we click on "Find train" after filling the above information, it returns a list of trains with the following information:

   (a) Arrival Time on Source.

   (b) Departure Time from Source.

   (c) Train Name

   (d) Train Number

   (e) Arrival Time on Destination

   (f) Arrival Date on Destination

3. The option "check availability" will redirect the respective train's information to the booking page.
4. On the booking page, the currently available seats of all types will be shown. Choose whichever type of seat you want to book (2AC, 2S, 3AC, SL etc). "Book" option will redirect the user to a new page, where user needs to fill the following details:

   (a) Name

   (b) Age

   (c) Gender

   (d) Email

   (e) Mobile

   (f) Seat

5. If user wants to add more passengers, click on "Add Ticket". And fill the details of the new passenger. Finally, click on "Book Ticket".

6. Finally, a ticket will be displayed to the user with a unique PNR no. and all the details filled by the user.
7. For PNR Enquiry, user need to click on 'PNR enquiry' on homepage and enter their PNR no. It will again show the ticket will all details.
8. To cancel a pre-booked ticket. User can go to 'Cancel Ticket' and cancel that ticket by entering their PNR no. and Name.

System View:

1. To create a dropdown of all stations while searching a train, we are calling a query that returns the name of all stations.
2. For given source, date, destination, the query is returning only those trains in which atleast one seat in atleast one class is empty.
3. View total-seats-available is created to check the total seats present for every train in every class. These seats can be booked or not booked.
4. Indexes are created on pnr no in pnr table, combined index on train number, date and coach number in pnr table because these searches are used very frequently for pnr enquiries, pnr changes and checking available seats respectively.
5. Index is also created on class in coach table because the view total-seats-available uses this search query and is used frequently.
6. At last an index is created on station and train in schedule because this relation is also used extensively.

Queries and sample run time:

1. extract all station names.

```
SELECT name from stations;
```

   Running Time : 4.697 ms

2. Train information for given source, destination and date:

```
SELECT s1.arrival, s1.departure, s1.train_name, s1.train_number, s2.arrival, \
TO_DATE('{date}','YYYY-MM-DD') + s2.day - s1.day AS arrival_date \
    FROM schedules as s1, \
        schedules as s2 \
    where s1.station_name = '{src}' \
    AND s2.station_name = '{dest}'  \
    AND s1.train_number = s2.train_number  \
    AND (s2.day > s1.day OR (s2.day = s1.day AND s2.arrival >= s1.departure)) \
    AND 0 < ( \
            SELECT (select sum(seats_available) from total_seats_available where \
            train_id = s1.train_number) - \
            ( \
```

```
                    select count(*) from PNR  \
                    where date ='{date}' \
                    AND train_number = s1.train_number \
                    AND delete=0 \
                ) \
            ) \
        ORDER BY s1.arrival;
```

Running Time: (src: JAIPUR, dest: AJMER JN, date: 2021-04-04) : 21.476 ms

3. Returns information about the remaining seats and their distribution in different classes for given train details.

```
SELECT s1.arrival AS arrival_src, s1.departure AS dept_src, s1.train_name, s1.train_number, \
s2.arrival AS arrival_dest, ts.class,ts.seats_available - COALESCE(pnr.count, 0) seats, \
TO_DATE('{date}','YYYY-MM-DD') + s2.day - s1.day AS arrival_date \
FROM schedules AS s1, \
schedules AS s2, \
total_seats_available AS ts \
LEFT JOIN \
( \
    SELECT coach.class, PNR.train_number, count(*) FROM PNR, coach \
    WHERE DATE ='{date}' \
    AND coach.coach_name = PNR.coach_no \
    AND PNR.delete = 0 \
    GROUP BY coach.class, PNR.train_number \
) AS pnr \
ON (pnr.train_number = ts.train_id \
    AND pnr.class = ts.class) \
WHERE s1.station_name = '{src}' \
AND s2.station_name = '{dest}'  \
AND s1.train_number = s2.train_number \
AND (s2.day > s1.day OR (s2.day = s1.day AND s2.arrival >= s1.departure)) \
AND s1.train_number = '{train_number}' \
AND ts.train_id = s1.train_number \
ORDER BY s1.arrival;
```

Running Time: (src: JAIPUR, dest: AJMER JN, date: 2021-04-04, train_number : 12958) : 2.106 ms

4. Returns type of all coaches in a train class:

```
SELECT distinct coach_type from coach where class = '{train_class}'
```

Running Time: (train_class: 2S): 0.716 ms

5. Returns coach names, and total seats in corresponding coaches for given train number:

```
SELECT coach_name, total_seats from coach where class = '{train_class}';
```

Running Time: (train_class : 2S) : 0.602 ms

6. Returns the seats that has already booked for given date and train_number.

```
SELECT PNR.coach_no, PNR.seat_no from PNR, coach where \
PNR.train_number = '{train_number}' AND PNR.delete = 0 \
and coach.class = '{train_class}' and \
coach.coach_name = PNR.coach_no and PNR.date = '{date}';
```

Running Time: (train_number = 12958, train_class: 2S, date: 2020-04-04): 0.737 ms

7. returns the newest assigned PNR no. (to create a new PNR)

```
SELECT pnr_no FROM pnr GROUP BY pnr_no ORDER BY pnr_no DESC LIMIT 1;
```

Running Time: 1.042 ms

8. Generate a new ticket by adding a new entry in PNR table.

```
INSERT INTO pnr VALUES ('{pnr_number}', '{train_number}', '{date}', '{seat_list[i][0]}', \
{seat_list[i][1]}, '{seat_list[i][2]}', '{name[i]}', {age[i]}, '{gender[i]}',\
'{mobile[i]}', '{email[i]}', '{src}', '{dest}', 0);
```

Running Time : 5.285 ms
Sample: ('0000000001', '12958', '2021-04-04', 'A1', 14, 'Lower', 'Manoj', 20, 'Male', '9876543210', 'cs5180411@iitd.ac.in', 'JAIPUR', 'AJMER JN', 0)

9. PNR Enquiry:

```
SELECT name, age, gender, email, mobile, seat_no, coach_no, birth_type, \
date, src, dest, delete as status, train_number FROM \
PNR where PNR_no = '{pnr}';
```

Running Time: (pnr = 0000000001) : 0.903 ms

10. Cancel Ticket:

```
UPDATE pnr SET delete = 1 WHERE pnr_no = '{pnr}';
```

Running Time: (pnr = 0000000002) : 2.68 ms

11. View total-seats-available:

```
CREATE VIEW total_seats_available AS
SELECT trains.number as train_id, coach.class as class, SUM(coach.total_seats) as seats_ava
FROM trains, coach
WHERE coach.class = '2S'
OR (coach.class = '1AC' AND trains.first_ac = 1)
OR (coach.class = '2AC' AND trains.second_ac = 1)
```

```
OR (coach.class = '3AC' AND trains.third_ac = 1)
OR (coach.class = 'SL' AND trains.sleeper = 1)
OR (coach.class = 'FC' AND trains.first_class = 1)
OR (coach.class IN ('CC', 'EC') AND trains.chair_car = 1)
GROUP BY trains.number, coach.class;
```

Running Time: 70.68 ms