

# **NEXTRAIL**

*Replacing IRCTC sem by sem*

**DBMS | CSE 202**

**Group 23**

**Aadit Kant Jha | Abhik S Basu | Mohit Jain | Sohum Sikdar**

# Table of Contents

- Project Scope
- Relational Schema
- Views and Grants
- SQL Queries
- Indexing
- Triggers
- Project Snapshots

# Project Scope

The Indian Railways has been an integral part of the Indian Economy since it's dawn. The whole ecosystem of Railways generates tons of data daily which needs to be structured and queried properly for a better user experience in terms of both comfort, easy, and speed and for better RnD for further development of such an integral part of the out country. This is where we come in with NEXTRAIL.

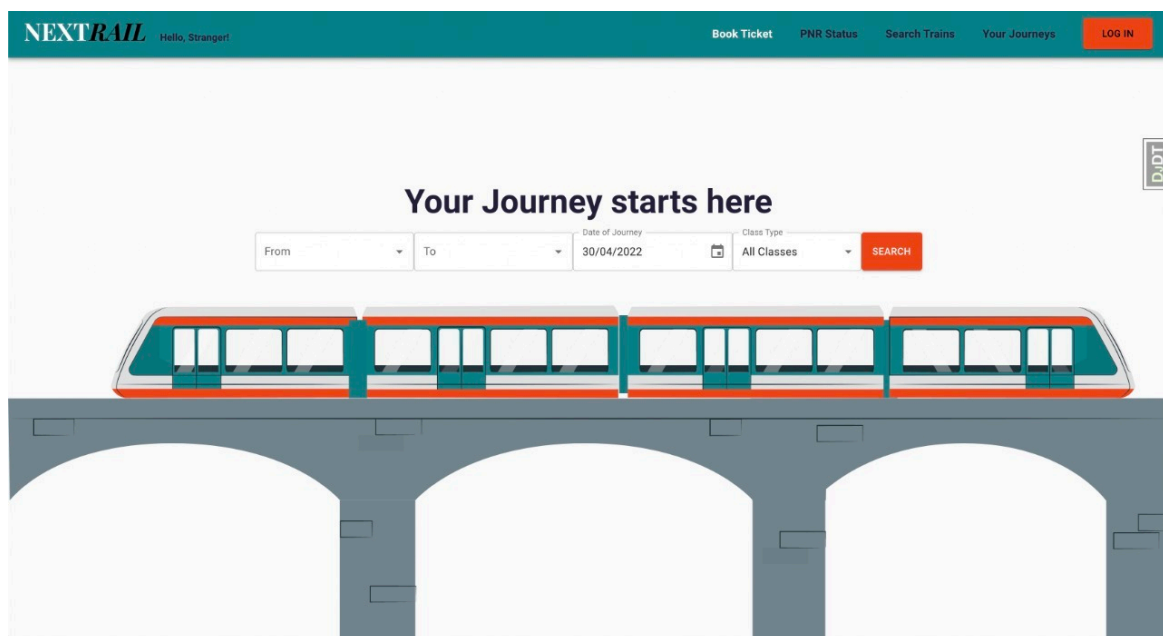
The user gets to book tickets with ease while choosing whatever class they wish to book while paying with their choice of payment method securely, choose their choice of meals, cancel current tickets, view previous tickets, view their current tickets status and get technical help whenever they require.

The tech-team can with ease update the schedules of existing trains, add new trains or delete existing trains. Further the workforce on field like the catering services would be able to fetch passenger preferences for better productivity.

Further, all the data that we collect and generate is kept in a structured manner that can be used by RnD teams for research purposes. During the midsem evaluation, the feedback given to our entities, relationships and ER diagram has helped us into making the final projects. While there were not any major issues in our midsem

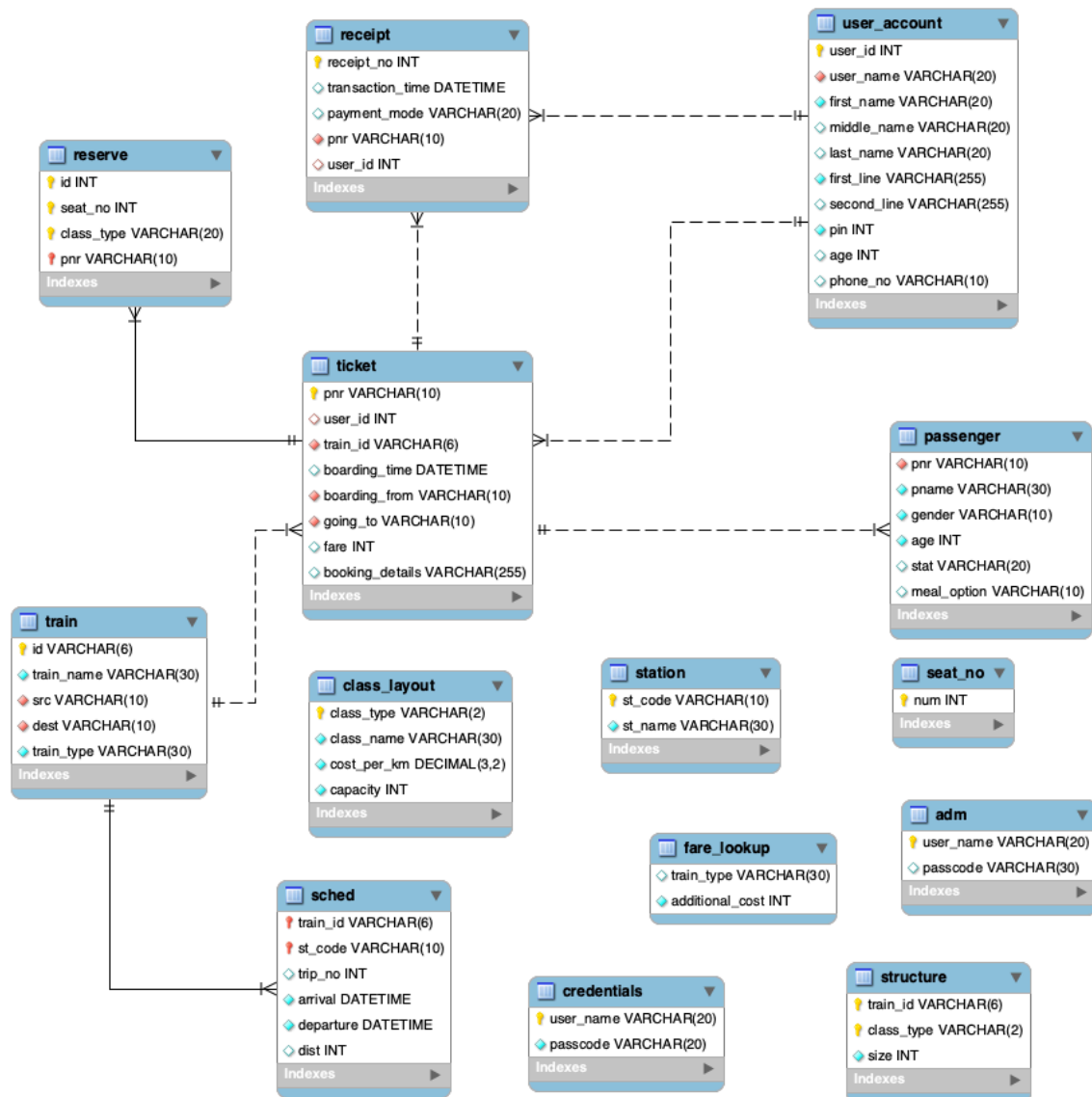
project as such. However we did face one shortcoming which was that we had not implemented any views in our code. As a result of which we have made sure to include views this time.

Furthermore, for our project we have made a user interface as well which is trying to model the real world IRCTC as much as possible. For this we have created a full stack project which involves usage of **React JS** and the **Material UI** framework on the front end side. And as far as the back end is concerned we used **Django** and queries were made using the **MySQL** workbench.



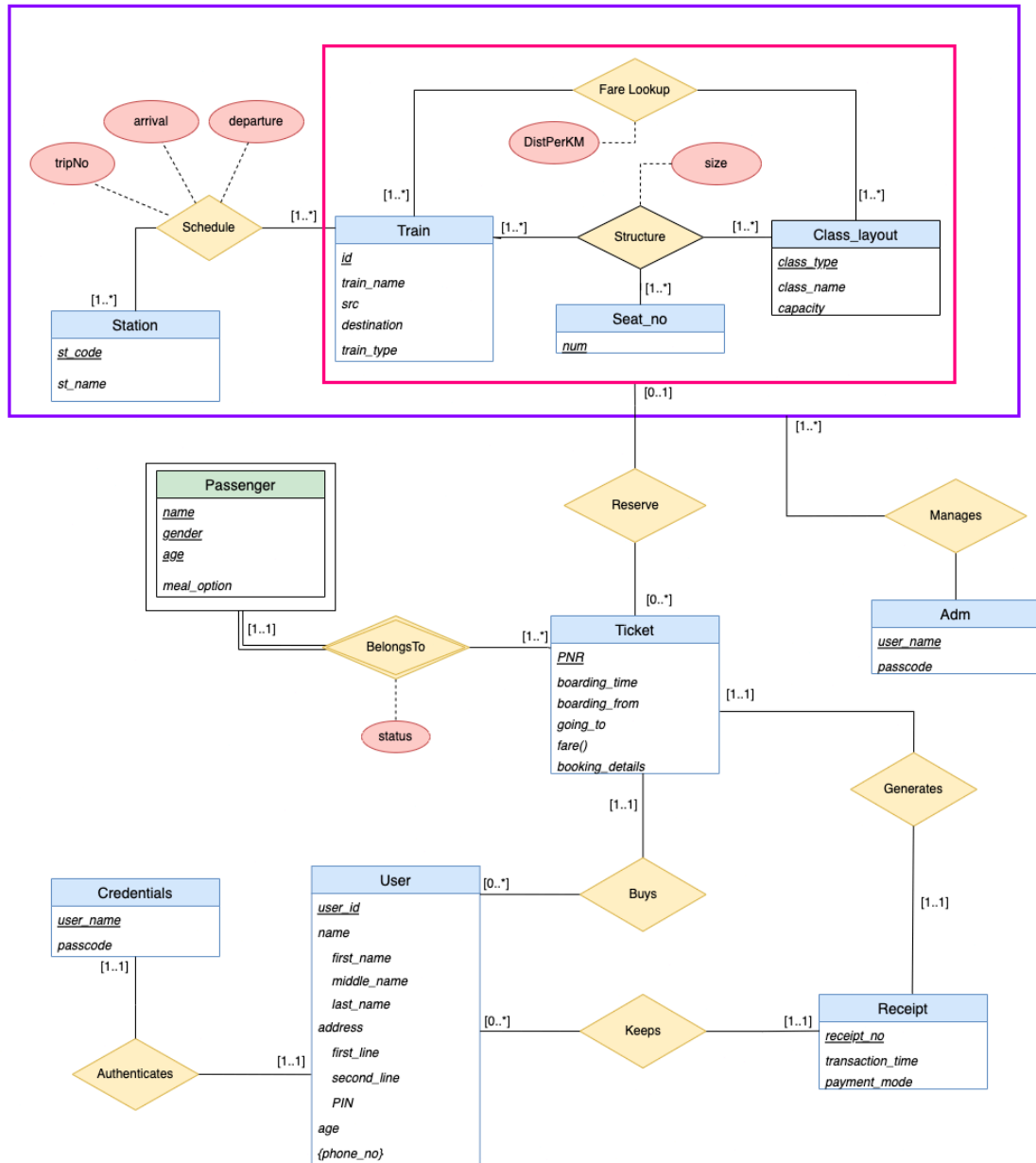
# Relational Schema

The following is our relational schema diagram



The following is our entity relationship diagram

### ER DIAGRAM OF NEXTRAIL



All other valid information on our stakeholders, entities, relations and constraints can be found in our midsem evaluation report ([https://github.com/MohitJain617/Railway-Reservation-System/blob/main/NEXTRAIL\\_ProjectReport.pdf](https://github.com/MohitJain617/Railway-Reservation-System/blob/main/NEXTRAIL_ProjectReport.pdf))

# Views and Grants

## Views

```
drop view if exists waiting_list_count;
CREATE VIEW waiting_list_count as
SELECT count(*) as WL, T.train_no, T.week_no, T.trip_no,
P.class_type as class_type
FROM passenger as P, ticket as T, receipt as R
WHERE P.pnr = T.pnr
      AND P.pnr = R.pnr
      AND P.stat = 'WL'
GROUP BY T.train_no, T.week_no, T.trip_no, P.class_type;
```

### **-- Waiting List View with ranks**

```
drop view if exists waiting_list;
CREATE VIEW waiting_list as
SELECT dense_rank() over (order by R.transaction_time) as
priority, P.pid, T.pnr, T.train_no, T.boarding_from,
T.going_to, T.week_no, T.trip_no, P.class_type
FROM passenger as P, ticket as T, receipt as R
WHERE P.pnr = T.pnr
      AND P.pnr = R.pnr
      AND P.stat = 'WL'
ORDER by priority,pid;
```

### **-- displays ticket details with everything from pnr to fare, dist, actual time and other stuff**

```
drop view if exists ticket_view;
create view ticket_view as
select T.pnr, T.train_no,
```

```

        (select train_name from train as T2 where T2.id =
T.train_no) as train_name,
        ((select dist from time_table as T2 where T2.train_no =
T.train_no and T2.st_code = T.going_to)-
        (select dist from time_table as T2 where T2.train_no =
T.train_no and T2.st_code = T.boarding_from)) as dist,
        TIMESTAMP(Date_add(get_daytime(week_no,trip_no-1)
,
        INTERVAL ((select day_no from time_table as TT
where TT.train_no = T.train_no and TT.st_code =
T.boarding_from)-1) day) ,
        (select departure from time_table as T2 where
T2.train_no = T.train_no and T2.st_code = T.boarding_from))
as srctime,
        TIMESTAMP(Date_add(get_daytime(week_no,trip_no-1)
,
        INTERVAL ((select day_no from time_table as TT
where TT.train_no = T.train_no and TT.st_code =
T.going_to)-1) day),
        (select arrival from time_table as T2 where T2.train_no
= T.train_no and T2.st_code = T.going_to)) as desttime,
        T.boarding_from, T.going_to, T.fare
from ticket as T;

```

drop view if exists seats\_view;

create view seats\_view as

```

SELECT S.train_no,
        S.class_type as coach, SN2.num as coach_no,
SN.num as seat_no FROM struct AS S,
        class_layout as C, seat_no AS SN2, seat_no as SN

```



```
WHERE S.class_type = C.class_type AND SN2.num <=
S.size
AND SN.num <= C.capacity;
```

```
drop view if exists stations_view;
create view stations_view as
select st_code,st_name,arrival,departure,dist,day_no
from time_table natural join station
order by dist ;
```

## Grants

```
CREATE ROLE 'Admins'@'localhost',
'Passenger'@'localhost', 'Financials'@'localhost',
'Ministry'@'localhost', 'Employees'@'localhost';
```

```
GRANT ALL ON *
TO 'Admins'@'localhost';
```

```
GRANT SELECT
ON waiting_list_count
TO 'Employees'@'localhost', 'Ministry'@'localhost',
'Passenger'@'localhost';
```

```
GRANT SELECT
ON waiting_list
TO 'Employees'@'localhost', 'Ministry'@'localhost',
'Passenger'@'localhost';
```

```
GRANT SELECT
ON ticket_view
```

```
TO  
'Passenger'@'localhost','Financials'@'localhost','Employees'  
@'localhost';
```

```
GRANT SELECT  
ON seats_view  
TO 'Passenger'@'localhost','Employees'@'localhost';
```

```
GRANT ALL ON seats_view  
TO 'Ministry'@'localhost';
```

```
GRANT SELECT  
ON stations_view  
TO 'Employees'@'localhost', 'Passenger'@'localhost';
```

```
GRANT SELECT  
ON station  
TO 'Employees'@'localhost', 'Passenger'@'localhost';
```

```
GRANT ALL  
ON station  
TO 'Ministry'@'localhost';
```

```
GRANT ALL  
ON train  
TO 'Ministry'@'localhost';
```

```
GRANT ALL  
ON time_table  
TO 'Ministry'@'localhost';
```

```
GRANT ALL  
ON sched  
TO 'Ministry'@'localhost';
```

```
GRANT ALL  
ON class_layout
```

TO 'Ministry'@'localhost';

GRANT ALL

ON seat\_no

TO 'Ministry'@'localhost';

GRANT SELECT

ON station

TO 'Employees'@'localhost', 'Passenger'@'localhost';

GRANT SELECT

ON sched

TO 'Employees'@'localhost','Passenger'@'localhost';

GRANT SELECT

ON train

TO 'Employees'@'localhost','Passenger'@'localhost';

GRANT SELECT

ON fare\_lookup

TO 'Employees'@'localhost','Passenger'@'localhost';

GRANT SELECT

ON class\_layout

TO 'Employees'@'localhost','Passenger'@'localhost';

GRANT SELECT

ON receipt

TO 'Financials'@'localhost';

GRANT UPDATE

ON receipt

TO 'Financials'@'localhost';

GRANT SELECT

ON fare\_lookup

```
TO 'Financials'@'localhost';
```

```
GRANT SELECT
```

```
ON class_layout
```

```
TO 'Financials'@'localhost';
```

# SQL Queries

## 1)Query for trains between two stations --

```
-- -----QUERY-----  
SELECT T.train_no, T.departure FROM time_table as T  
NATURAL JOIN sched as S  
WHERE T.st_code = @tempsrc  
      AND (T.day_no+S.trip_no-1) = @tempdayno  
      AND EXISTS (  
          SELECT * FROM time_table as T2 NATURAL JOIN  
sched as S2  
          WHERE S2.trip_no = S.trip_no  
                AND S2.train_no = S.train_no  
                AND (T.dist) < (T2.dist)  
                AND T2.st_code = @tempdest  
          )  
      AND (  
          (@classReq = FALSE) OR EXISTS (  
              SELECT * FROM struct as STR  
              WHERE STR.train_no = T.train_no  
                    AND STR.class_type = @classPref  
          )  
      )  
);  
-- -----
```

## 2)Query for trains between two stations sorted by Departure time at source --

```
-- -----QUERY-----  
SELECT T.train_no, T.departure FROM time_table as T  
NATURAL JOIN sched as S
```

```

WHERE T.st_code = @tempsrc
      AND (T.day_no+S.trip_no-1) = @tempdayno
      AND EXISTS (
          SELECT * FROM time_table as T2 NATURAL JOIN
sched as S2
          WHERE S2.trip_no = S.trip_no
            AND S2.train_no = S.train_no
            AND (T.dist) < (T2.dist)
            AND T2.st_code = @tempdest
        )
      AND (
          (@classReq = FALSE) OR EXISTS (
              SELECT * FROM struct as STR
              WHERE STR.train_no = T.train_no
              AND STR.class_type = @classPref
          )
      )
) ORDER BY T.departure;

```

-----

### **3) Query to find all seats given a train**

-----SETUP-----

```
SET @temptrain = '11123';
```

-----

-----QUERY-----

```

SELECT S.train_no, S.class_type as coach, SN2.num as
coach_no, SN.num as seat_no
FROM structure AS S, class_layout as C, seat_no AS SN2,
seat_no as SN
WHERE S.train_no = @temptrain
      AND S.class_type = C.class_type
      AND SN2.num <= S.size
      AND SN.num <= C.capacity

```

ORDER BY (coach\_no,seat\_no);

-----

**4)Query for available seats between two stations for a train and a given date, src, and dest**

-- -----QUERY-----

```
SELECT S.train_no, S.class_type as coach, SN2.num as
coach_no, SN.num as seat_no
FROM struct AS S, class_layout as C, seat_no AS SN2,
seat_no as SN
WHERE S.train_no = @temptrain
      AND S.class_type = C.class_type
      AND SN2.num <= S.size
      AND SN.num <= C.capacity
      AND NOT EXISTS (
          SELECT * FROM reserve as R, ticket as T,
passenger as P
          WHERE T.train_no = S.train_no AND R.class_type
= S.class_type AND R.coach_no = SN2.num AND R.seat_no
= SN.num
          AND R.pnr = T.pnr
          AND T.pnr = P.pnr AND P.stat='CNF'
          AND T.train_no = @temptrain
          AND T.trip_no = @tripno
          AND T.week_no = @tripweek
          AND NOT(
              (
                  (SELECT dist FROM time_table as TT1
                  WHERE TT1.train_no = T.train_no
                  AND TT1.st_code = @tempdest)
                  <=
```

```

        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no = T.train_no
        AND TT2.st_code = T.boarding_from)
    )
    OR
    (
        (SELECT dist FROM time_table as TT1
        WHERE TT1.train_no = T.train_no
        AND TT1.st_code = @tempsrc)
        >=
        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no = T.train_no
        AND TT2.st_code = T.going_to)
    )
)
);

```

-----

## 5) Calculate the waiting list queue for a given train

-- -----QUERY-----

```

SELECT count(*) as WL, W.class_type
FROM waiting_list as W
WHERE W.train_no = @trainNo
      AND W.week_no = @tripweek
      AND W.trip_no = @tripno
      AND NOT(
        (
          (SELECT dist FROM time_table as TT1
          WHERE TT1.train_no = W.train_no
          AND TT1.st_code = @tempdest)

```



```

        <=
        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no =W.train_no
        AND TT2.st_code = W.boarding_from)
    )
    OR
    (
        (SELECT dist FROM time_table as TT1
        WHERE TT1.train_no = W.train_no
        AND TT1.st_code = @tempsrc)
        >=
        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no = W.train_no
        AND TT2.st_code = W.going_to)
    )
) GROUP BY class_type;

```

## 6) Get the count of available seats on a specific train and class type

-- -----QUERY-----

```

SELECT count(*) as Avail, S.class_type as class_type
FROM struct AS S, class_layout as C, seat_no AS SN2,
seat_no as SN
WHERE S.train_no = @trainNo
    AND S.class_type = C.class_type
    AND SN2.num <= S.size
    AND SN.num <= C.capacity
    AND NOT EXISTS (
        SELECT * FROM reserve as R, ticket as T,
passenger as P

```

```

        WHERE T.train_no = S.train_no AND R.class_type
= S.class_type AND R.coach_no = SN2.num AND R.seat_no
= SN.num
        AND R.pnr = T.pnr
        AND T.pnr = P.pnr AND P.stat='CNF'
        AND T.train_no = @trainNo
        AND T.trip_no = @tripno
        AND T.week_no = @tripweek
        AND NOT(
            (
                (SELECT dist FROM time_table as TT1
                WHERE TT1.train_no = T.train_no
                AND TT1.st_code = @tempdest)
                <=
                (SELECT dist FROM time_table as TT2
                WHERE TT2.train_no = T.train_no
                AND TT2.st_code = T.boarding_from)
            )
            OR
            (
                (SELECT dist FROM time_table as TT1
                WHERE TT1.train_no = T.train_no
                AND TT1.st_code = @tempsrc)
                >=
                (SELECT dist FROM time_table as TT2
                WHERE TT2.train_no = T.train_no
                AND TT2.st_code = T.going_to)
            )
        )
    ) GROUP BY S.class_type;

```

-----

## **7) Calculating duration of time it takes to reach station (in seconds)**

```
-- -----QUERY-----  
SELECT T1.st_code,  
TIME_TO_SEC(TIMEDIFF(T1.arrival,T2.departure))  
+86400*(T1.day_no-T2.day_no) as seconds  
FROM time_table as T1, time_table as T2  
WHERE T1.train_no = @trainno  
AND T2.train_no = @trainno  
AND T2.dist = 0  
ORDER BY seconds;  
-- -----
```

## **8) Query to check all the passengers in waiting list for a specific train info, and given them seat if it's available now**

```
-- -----QUERY-----  
SELECT *  
FROM waiting_list as W  
WHERE W.train_no = @trainNo  
      AND W.week_no = @tripweek  
      AND W.trip_no = @tripno  
      AND W.class_type = @classtype  
  
      AND NOT EXISTS (  
          SELECT *  
          FROM waiting_list as W2  
          WHERE W.train_no = W2.train_no  
          AND W.trip_no = W2.trip_no  
          AND W.class_type = W2.class_type  
          AND ((W.priority > W2.priority) OR ((W.priority =  
W2.priority) AND (W.pid > W2.pid)) )
```

```

-- W2 lies between tempsrc and tempdest
AND(
    (
        (SELECT dist FROM time_table as TT1
        WHERE TT1.train_no = W2.train_no
        AND TT1.st_code = @tempsrc)
        <=
        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no = W2.train_no
        AND TT2.st_code = W2.boarding_from)
    )
    AND
    (
        (SELECT dist FROM time_table as TT1
        WHERE TT1.train_no = W2.train_no
        AND TT1.st_code = @tempdest)
        >=
        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no = W2.train_no
        AND TT2.st_code = W2.going_to)
    )
)
-- and intersection between W1 and W2
AND NOT(
    (
        (SELECT dist FROM time_table as TT1
        WHERE TT1.train_no = W2.train_no
        AND TT1.st_code = W2.going_to)
        <=
        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no =W.train_no

```

```

        AND TT2.st_code = W.boarding_from)
    )
    OR
    (
        (SELECT dist FROM time_table as TT1
        WHERE TT1.train_no = W2.train_no
        AND TT1.st_code = W2.boarding_from)
        >=
        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no = W.train_no
        AND TT2.st_code = W.going_to)
    )
)
-- W1 lies between tempsrc and tempdest
AND(
    (
        (SELECT dist FROM time_table as TT1
        WHERE TT1.train_no = W.train_no
        AND TT1.st_code = @tempsrc)
        <=
        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no = W.train_no
        AND TT2.st_code = W.boarding_from)
    )
    AND
    (
        (SELECT dist FROM time_table as TT1
        WHERE TT1.train_no = W.train_no
        AND TT1.st_code = @tempdest)
        >=

```

```

        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no = W.train_no
        AND TT2.st_code = W.going_to)
    )
)
;

```

## 9) Query for all tickets booked by a user along with additional values

-----QUERY-----

```

select T.pnr, T.train_no,
(select train_name from train as T2 where T2.id =
T.train_no) as train_name,
((select dist from time_table as T2 where T2.train_no =
T.train_no and T2.st_code = T.going_to)-
    (select dist from time_table as T2 where T2.train_no =
T.train_no and T2.st_code = T.boarding_from)) as dist,
TIMESTAMP(Date_add(get_daytime(week_no,trip_no-1),
    INTERVAL ((select day_no from time_table as TT
where TT.train_no = T.train_no and TT.st_code =
T.boarding_from)-1) day) ,
    (select departure from time_table as T2 where
T2.train_no = T.train_no and T2.st_code = T.boarding_from))
as srctime,
TIMESTAMP(Date_add(get_daytime(week_no,trip_no-1),
    INTERVAL ((select day_no from time_table as TT
where TT.train_no = T.train_no and TT.st_code =
T.going_to)-1) day),
    (select arrival from time_table as T2 where T2.train_no =
T.train_no and T2.st_code = T.going_to)) as desttime,
T.boarding_from, T.going_to, T.fare

```

```

from ticket as T
where T.username = "test2"
AND
(TIMESTAMP(Date_add(get_daytime(week_no,trip_no-1),
      INTERVAL ((select day_no from time_table as TT
where TT.train_no = T.train_no and TT.st_code =
T.going_to)-1) day),
      (select arrival from time_table as T2 where T2.train_no =
T.train_no and T2.st_code = T.going_to)) < "2022-04-24
03:50:00");
-----

```

**10) Given pid of passenger calculate the waiting list number if it's waiting listed**

```

-- -----QUERY-----
SELECT count(*) as WL
FROM waiting_list as W, waiting_list as W2
WHERE W.train_no = W2.train_no
      AND W.week_no = W2.week_no
      AND W.trip_no = W2.trip_no
      AND W.class_type = W2.class_type
      AND W2.pid = @ppid
      AND W2.pid <> W.pid
      AND ((W2.priority > W.priority) or ((W2.priority =
W.priority) and (W2.pid >= W.pid)))
      AND NOT(
      (
      (SELECT dist FROM time_table as TT1
WHERE TT1.train_no = W.train_no
AND TT1.st_code = W2.going_to)
      <=

```

```

        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no =W.train_no
        AND TT2.st_code = W.boarding_from)
    )
    OR
    (
        (SELECT dist FROM time_table as TT1
        WHERE TT1.train_no = W.train_no
        AND TT1.st_code = W2.boarding_from)
        >=
        (SELECT dist FROM time_table as TT2
        WHERE TT2.train_no = W.train_no
        AND TT2.st_code = W.going_to)
    )
);

```

-----

#### -- **Extras:**

-- Calculate fare

SET @pcnt = 1;

SET @dist = 1384;

SET @classType = 'A';

SET @trainno = '22210';

select @pcnt\*(@dist)\*(select distinct cost\_per\_km from  
class\_layout as C where C.class\_type=@classType) +  
(SELECT DISTINCT FL.additional\_cost FROM fare\_lookup  
as FL, train as T WHERE T.id=@trainno AND  
T.train\_type=FL.train\_type) as fare;



# Indexing

```
CREATE INDEX station_index ON station (st_name);
CREATE INDEX trainname_index ON train (train_name);
CREATE INDEX traintype_index ON train (train_type);
CREATE INDEX ticket_index ON ticket (trip_no, week_no);
CREATE INDEX ticket_fare ON ticket(fare);
CREATE INDEX receipt_index ON receipt (pnr);
CREATE INDEX passenger_index ON passenger(pnr);
CREATE INDEX class_layout_index ON class_layout
(class_name);
CREATE INDEX time_table_index ON time_table (train_no);
```

# Triggers

## **find additional cost given a train type**

```
drop trigger if exists additional_cost;
-- TRIGGER FOR ADDITIONAL FARE COST
DELIMITER $$
CREATE TRIGGER additional_cost
BEFORE INSERT ON ticket FOR EACH ROW
BEGIN
    SET NEW.fare = (SELECT DISTINCT FL.additional_cost
FROM fare_lookup as FL, train as T WHERE
T.id=NEW.train_no AND T.train_type=FL.train_type);
END;
$$
DELIMITER ;

drop trigger if exists fare_update;
```

```

-- TRIGGER FOR UPDATION IN FARE COST AS
PASSENGERS ARE ADDED
DELIMITER $$
CREATE TRIGGER fare_update
AFTER INSERT ON passenger FOR EACH ROW
begin
    -- CALCULATE DISTANCE
    SET @distsrc = (SELECT dist FROM time_table as
T, ticket as TC
                    WHERE TC.pnr = NEW.pnr
                    AND T.train_no = TC.train_no
                    AND T.st_code =
TC.boarding_from) ;

    SET @distdest = (SELECT dist FROM time_table
as T, ticket as TC
                    WHERE TC.pnr = NEW.pnr
                    AND T.train_no = TC.train_no
                    AND T.st_code = TC.going_to) ;

    UPDATE ticket
    SET fare = fare+(@distdest-@distsrc)*(select
distinct cost_per_km from class_layout as C where
C.class_type=NEW.class_type)
    WHERE pnr = NEW.pnr;
END;
$$
DELIMITER ;

```

### **Trigger for auto generation of receipt**

```
drop trigger if exists gen_receipt;
```

```

DELIMITER $$
CREATE TRIGGER gen_receipt
AFTER INSERT ON ticket FOR EACH ROW
begin
    SET @lastval = (select max(receipt_no) from
receipt);
    SET @lastval = ifnull(@lastval,0);
    INSERT INTO receipt VALUES
    (@lastval+1,now(),'Pending',NEW.pnr, NEW.username);
END;
$$
DELIMITER ;

```

### **Trigger for assigning seat to a passenger**

```

drop trigger if exists book_seat_if_avail;
DELIMITER $$
CREATE TRIGGER book_seat_if_avail
BEFORE INSERT ON passenger FOR EACH ROW
begin
    SET @seatavail = false;

    SET @temptrain = (SELECT train_no from ticket where
pnr = NEW.pnr);
    SET @tempsrc = (SELECT boarding_from from ticket
where pnr = NEW.pnr);
    SET @tempdest = (SELECT going_to from ticket
where pnr = NEW.pnr);

    set @tripno = (SELECT trip_no from ticket where pnr =
NEW.pnr);

```

```
set @tripweek = (SELECT week_no from ticket where
pnr = NEW.pnr);
```

```
set @tempclass = NEW.class_type;
```

```
-- different variables
SET @coachno = null;
SET @seatno = null;
SET @coachtype = null;
SELECT S.class_type, SN2.num, SN.num
INTO @coachtype, @coachno, @seatno
FROM struct AS S, class_layout as C, seat_no AS
SN2, seat_no as SN
WHERE S.train_no = @temptrain
      AND S.class_type = @tempclass
      AND S.class_type = C.class_type
      AND SN2.num <= S.size
      AND SN.num <= C.capacity
      AND NOT EXISTS (
          SELECT * FROM reserve as R, ticket as T,
passenger as P
          WHERE T.train_no = S.train_no AND
R.class_type = S.class_type AND R.coach_no = SN2.num
AND R.seat_no = SN.num
          AND R.pnr = T.pnr
          AND T.pnr = P.pnr AND P.stat='CNF'
          AND T.train_no = @temptrain
          AND T.trip_no = @tripno
          AND T.week_no = @tripweek
          AND NOT(
              (
```

```

TT1      (SELECT dist FROM time_table as

WHERE TT1.train_no = T.train_no
AND TT1.st_code = @tempdest)
<=
TT2      (SELECT dist FROM time_table as

WHERE TT2.train_no = T.train_no
AND TT2.st_code =
T.boarding_from)
)
OR
(
TT1      (SELECT dist FROM time_table as

WHERE TT1.train_no = T.train_no
AND TT1.st_code = @tempsrc)
>=
TT2      (SELECT dist FROM time_table as

WHERE TT2.train_no = T.train_no
AND TT2.st_code = T.going_to)
)
)
) LIMIT 1;

IF (@seatno IS NULL) then
    SET NEW.stat = 'WL';
ELSE
    SET NEW.stat = 'CNF';
    INSERT INTO reserve VALUES (@coachno,
@seatno, @coachtype, NEW.pnr);

```

END IF;

END;

\$\$

DELIMITER ;

# Project Snapshots

**NEXTRAIL** Hello, Stranger! [Book Ticket](#) [PNR Status](#) [Search Trains](#) [Your Journeys](#) [LOG IN](#)

### Your Journey starts here


From

To

Date of Journey

Class Type

[SEARCH](#)



DJDT

**NEXTRAIL** Hello, Stranger! [Book Ticket](#) [PNR Status](#) [Search Trains](#) [Your Journeys](#) [LOG IN](#)

## Search Trains

Enter Train No.

[SEARCH](#)

### New Delhi Mumbai Central DURONTO – 22210 New Delhi (NDLS) To Mumbai Central (MMCT)

New Delhi (NDLS) - Mumbai Central (MMCT)

#### Train Details

Train Code	Train Type	Classes	Duration	Distance	Stops
22210	DUR	A,B,H	17 hrs 40 mins	1384	5

#### Stations

Station Name	Code	Day	Arrival	Departure	Distance
New Delhi	NDLS	1	22:10:00	22:10:00	0
Kota Jn	KOTA	2	03:40:00	03:50:00	465
Ratlam Jn	RTM	2	07:00:00	07:05:00	731
Vadodara Jn	BRC	2	10:33:00	10:43:00	993
Mumbai Central	MMCT	2	15:50:00	15:50:00	1384

DJDT

Hello, Stranger!

[Book Ticket](#)
[PNR Status](#)
[Search Trains](#)
[Your Journeys](#)
[LOG IN](#)

# PNR Status

Enter PNR No:  
4309403590

SEARCH

New Delhi Mumbai  
Central DURONTO  
NDLS  
22:10  
2022-04-23

→
  
17h 40m

1384 km  
22210  
MMCT  
15:50  
2022-04-24

PNR No. : 4309403590
Fare: ₹5482

SNo.	Passenger Name	Current Status	Coach No.	Seat No.	Meal Option
0	Purjit J	CNF	A2	1	-
1	Mohit J	CNF	A2	3	-

Hello, Stranger!

[Book Ticket](#)
[PNR Status](#)
[Search Trains](#)
[Your Journeys](#)
[LOG IN](#)

# Your Journey starts here

From  
Muzaffarpur Jn (...)

To  
Barauni Jn (BJU)

Date of Journey  
03/05/2022

Class Type  
AC 2 Tier

Passengers  
2 Passengers

SEARCH

Barauni Jn Gwalior Jn  
MAIL  
MFP  
09:25  
2022-05-03

→
  
3 hrs 25 mins

103km  
11123  
BJU  
12:50  
2022-05-03

AC 2 Tier (A)  
AVAILABLE : 54  
₹ 420

AC 3 Tier (B)  
AVAILABLE : 192  
₹ 359

Sleeper Class (S)  
AVAILABLE : 720  
₹ 163

Book Now