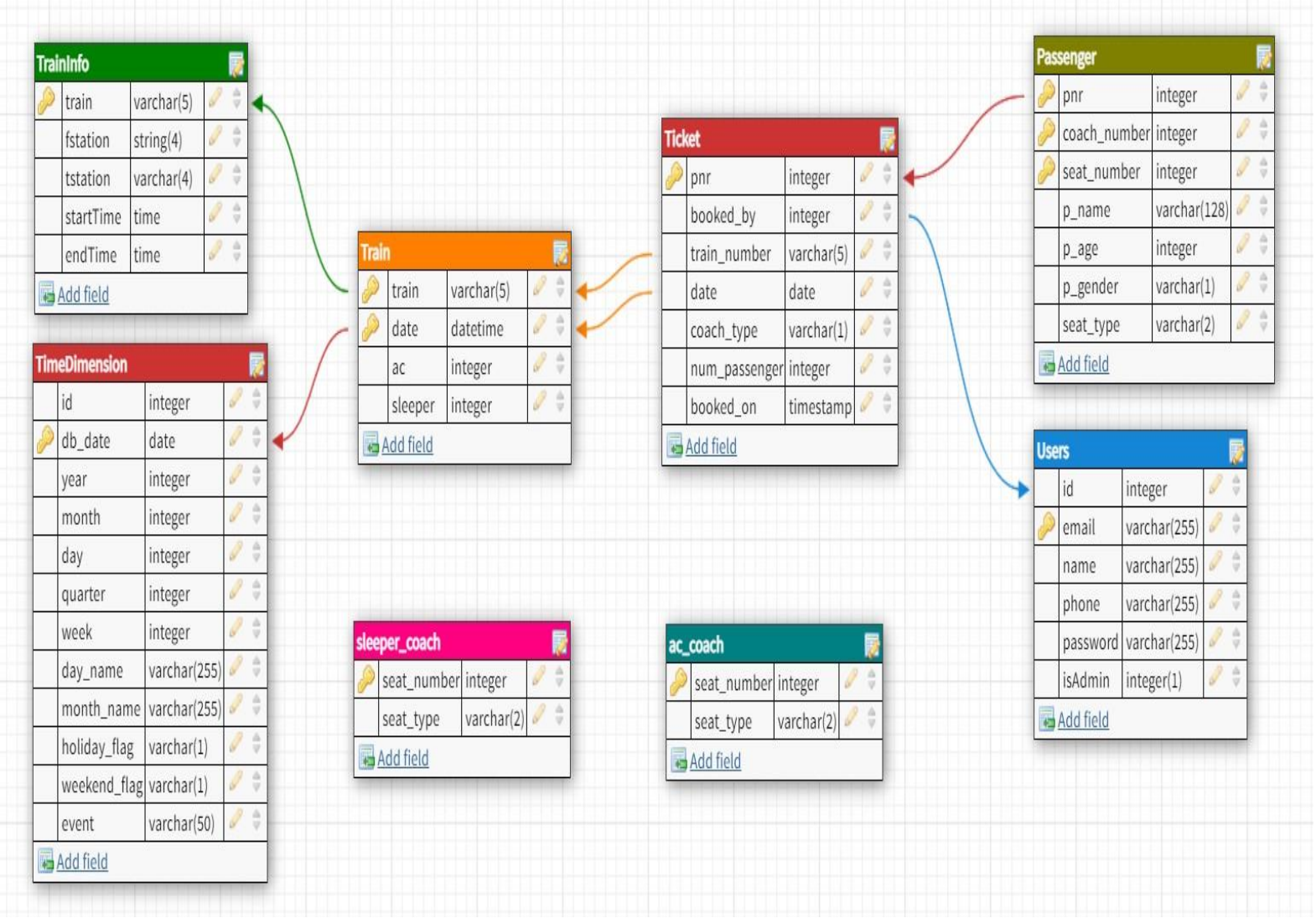
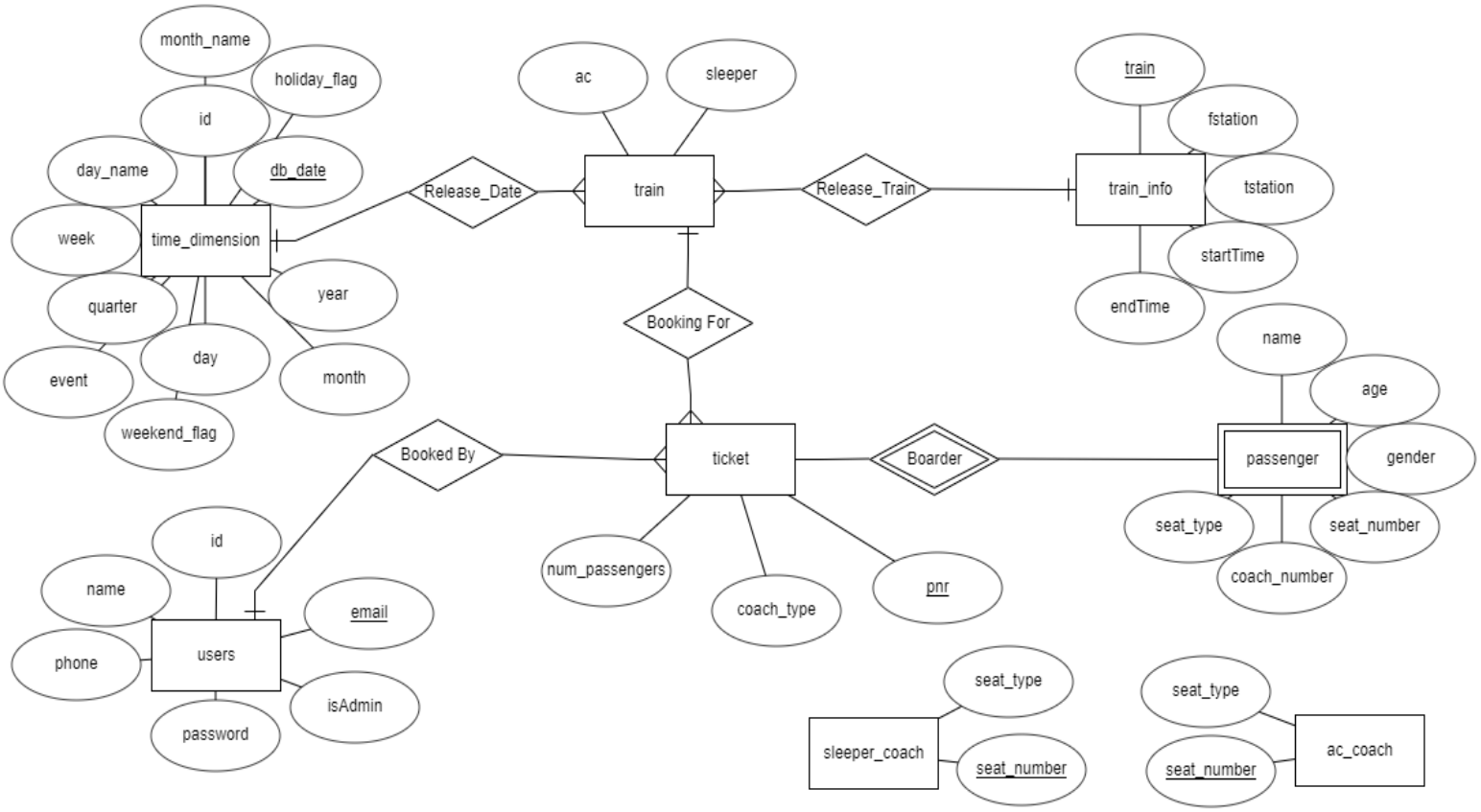
CS301 Database Systems
Course Project -> Railway Reservation System

Database Schema



Entity Relationship Diagram:



Triggers and Stored Procedures Used:

- ***(trigger)Insert_ticket:*** This trigger fires before an insert on the ticket table. Its function is to check whether the new ticket being inserted into the table has a seat requirement less than or equal to the number of available seats according to train capacity. The Train Capacity is retrieved from the train table and the number of seats filled is retrieved from the ticket table by conditioning on the train_number, date, and the coach_type of the new ticket we are trying to insert. If the required number of seats is not available, the insert operation is blocked by an SQL Exception. It is implemented as SQL Trigger.
- ***(trigger)insert_passenger***: This trigger makes sure that the current passenger being inserted should not exceed the maximum value of passenger allowed by the corresponding PNR in the ticket table. It is implemented as SQL Trigger.
-

```
PROCEDURE
    InsertPassenger(
                        IN pnr INT,
                        IN coach_type CHAR(1),
                        IN p_name VARCHAR(128),
                        IN p_age INT,
                        IN p_gender CHAR(1)
                        )
```

This stored procedure is used to add a passenger into the table passenger.
Whenever a new ticket is booked, the availability of seats is checked by insert_ticket trigger. On being successful, a new ticket is added to the ticket table with a unique pnr number. Post this operation, the procedure "insertPassenger" is called for every passenger.
A new seat number and coach number is allocated which has not been booked until then. Seat Type is then inferred from the coach_table (ac_coach or sleeper_coach). All the information is then added into the passenger table.
Note: Seats are allocated in a sequential manner and are uniquely identified by a combination of seat_number and coach_number.

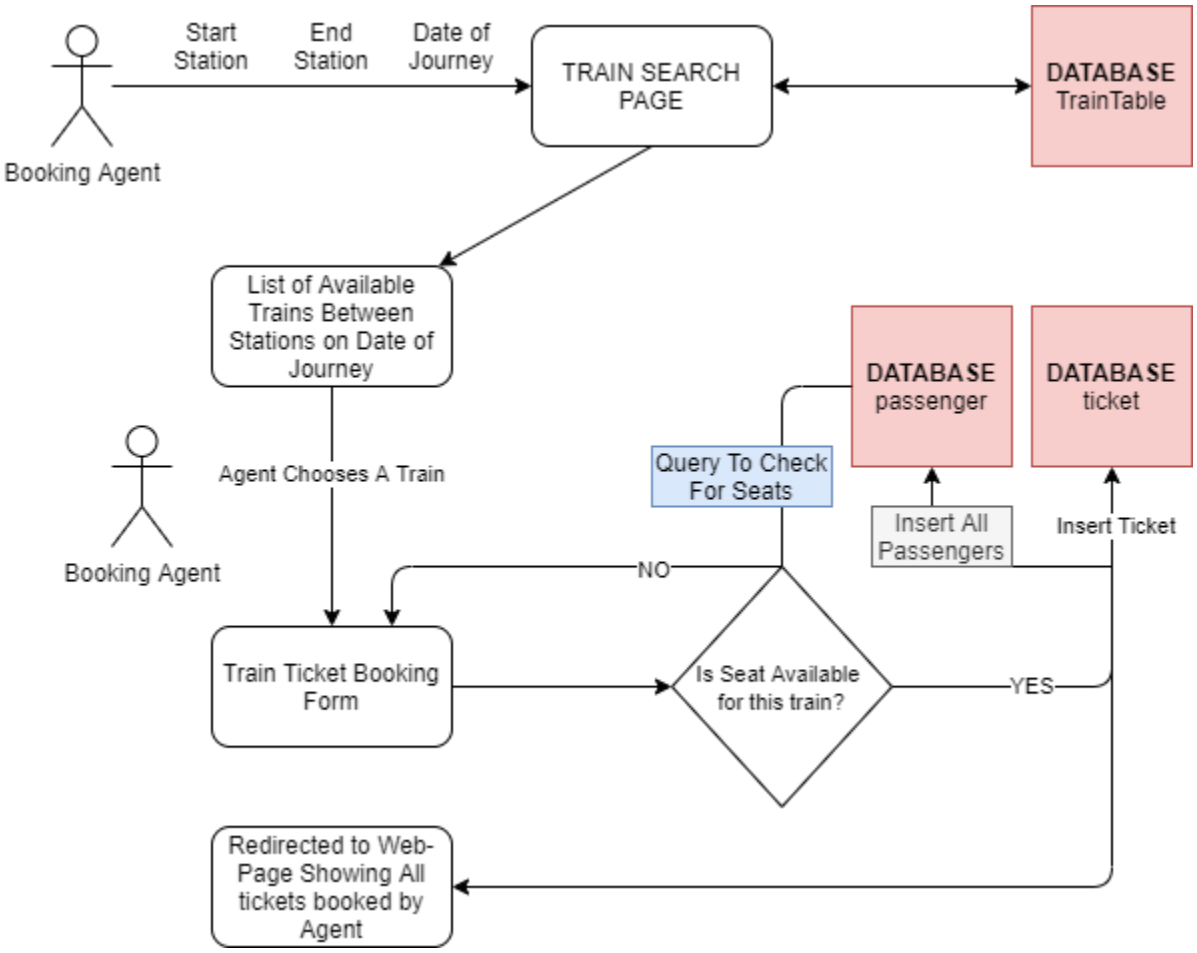
- `PROCEDURE fill_date_dimension(IN startdate DATE,IN stopdate DATE)`
  This procedure is used to populate the time_dimension table. It takes start_date and stop_date and inserts all the corresponding entries into the time_dimension table.

- `PROCEDURE `createAdmin` (IN `name1` VARCHAR(20), IN `email1` VARCHAR(255), IN `phone1` VARCHAR(15), IN `password1` VARCHAR(255)) BE`
  This procedure is used to create a new user with Administrator Privileges.

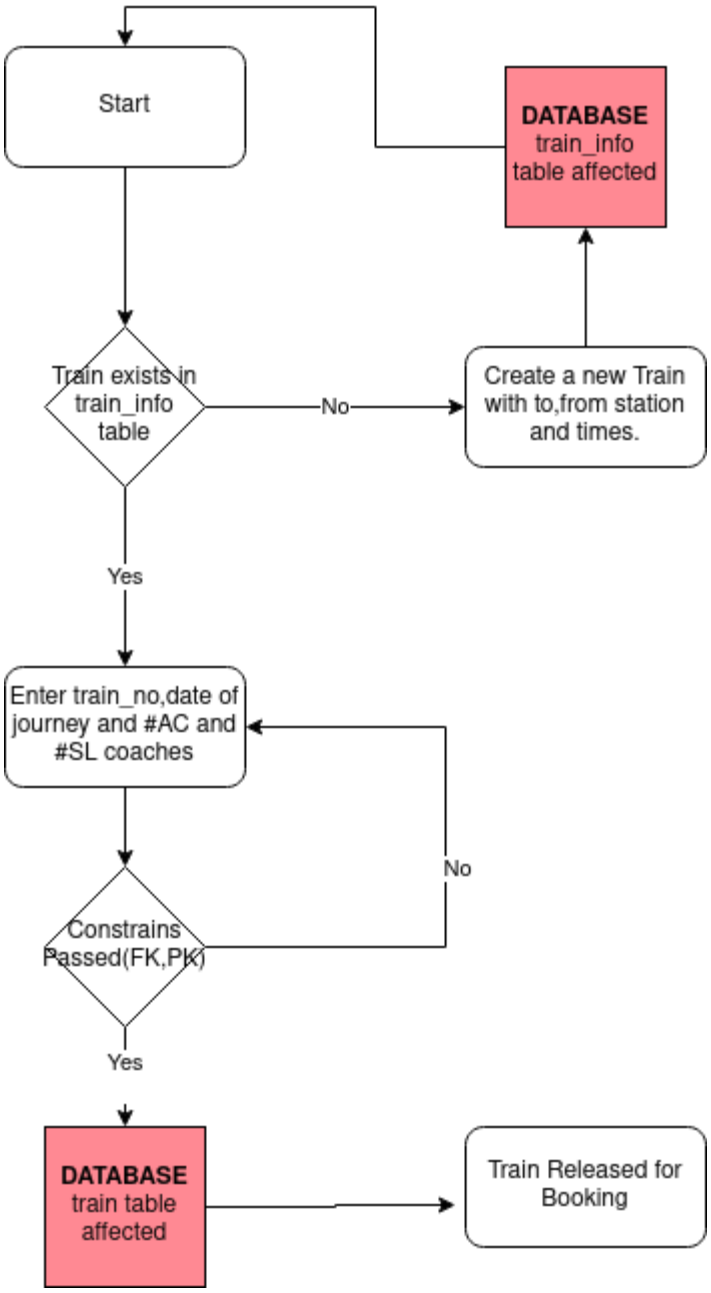
Other Triggers which maintain Admin Constraints(Only admin can create and release trains) are implemented as checks at the Frontend.
Triggers which maintain date constraints(booking can be made

between next date and 2 months from today),(Admin can release train between 2 months and 6 months from now) are implemented as part of the frontend.
Constraints requiring NON NULL Values are also implemented as frontend.

FLOW-CHART TRAIN BOOKING



FLOW CHART FOR RELEASING TRAINS


Start
DATABASE
train_info
table affected
Train exists in
train_info
table
No
Create a new Train
with to,from station
and times.
Yes
Enter train_no,date of
journey and #AC and
#SL coaches
No
Constrains
Passed(FK,PK)
Yes
DATABASE
train table
affected
Train Released for
Booking