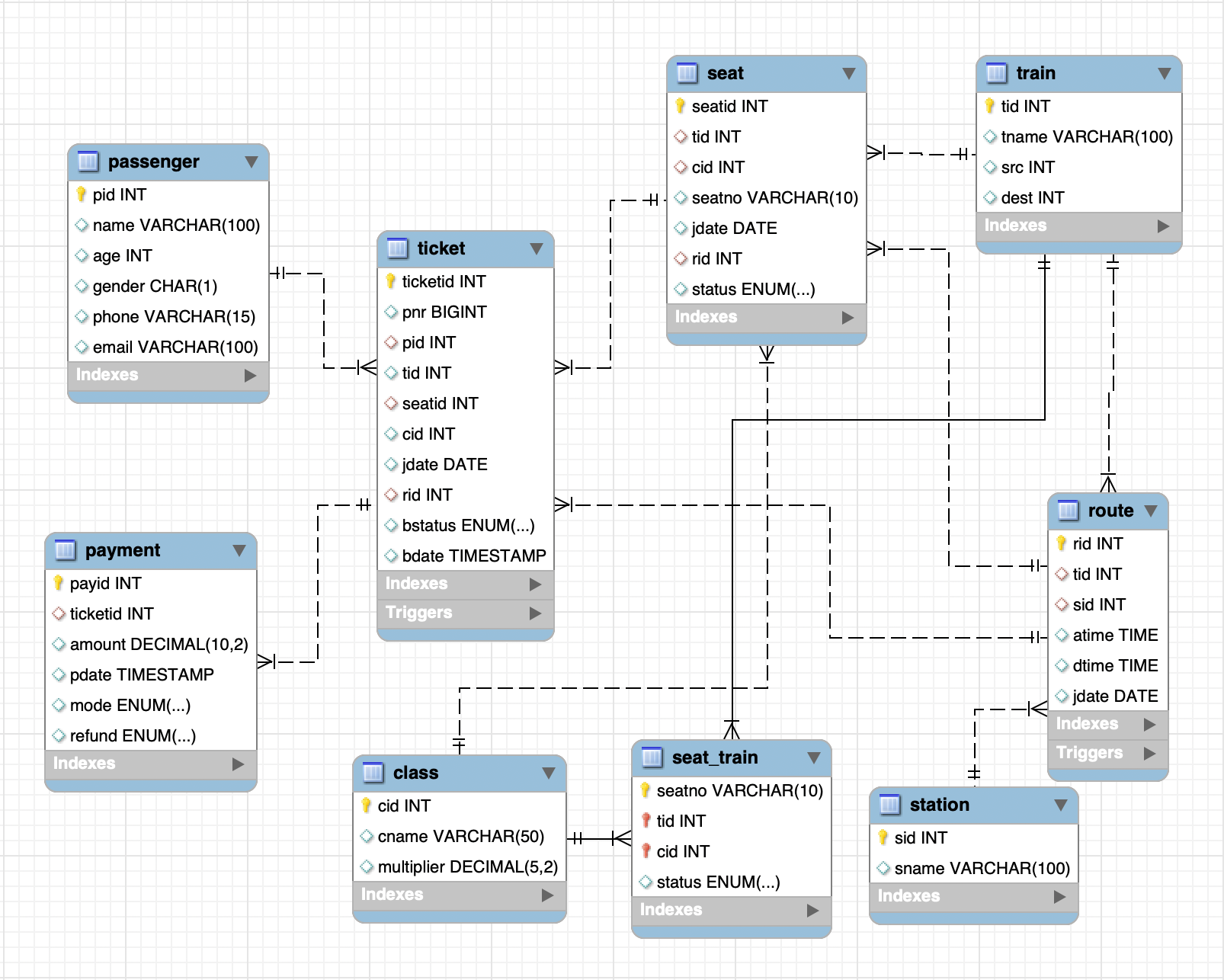
CS2202 DBMS PROJECT

INDIAN RAILWAYS DATABASE



**ER DIAGRAM**

# **TABLES**

create table passenger (

pid int auto\_increment primary key,

name varchar(100),

age int,

gender char(1),

phone varchar(15),

email varchar(100)

);

create table train (

tid int auto\_increment primary key,

tname varchar(100),

src int,

dest int

);

create table station (

sid int auto\_increment primary key,

sname varchar(100)

);

create table route (

rid int auto\_increment primary key,

tid int,

sid int,

atime time,

dtime time,

jdate date,

foreign key (tid) references train(tid),

foreign key (sid) references station(sid),

UNIQUE (jdate , tid , sid , atime)

);

create table class (

cid int auto\_increment primary key,

cname varchar(50),

multiplier decimal(5,2)

);

*-- seat will be dynamiclly maintained*

create table seat (

seatid int auto\_increment primary key,

tid int,

cid int,

seatno varchar(10),

jdate date,

rid int,

status enum('available', 'booked', 'rac', 'waitlist'),

foreign key (rid) references route(rid),

foreign key (tid) references train(tid),

foreign key (cid) references class(cid),

UNIQUE (seatno, rid , tid , cid)

);

create table seat\_train(

seatno varchar(10),

tid int,

cid int,

status enum('available', 'rac' ),

primary key (seatno, tid, cid),

foreign key (tid) references train(tid),

foreign key (cid) references class(cid)

);

create table ticket (

ticketid int auto\_increment primary key,

pnr bigint unique,

pid int,

tid int,

seatid int,

cid int,

jdate date,

rid int,

bstatus enum('confirmed', 'rac', 'waitlist', 'cancelled'),

bdate timestamp default current\_timestamp,

foreign key (pid) references passenger(pid),

foreign key (seatid) references seat(seatid ),

foreign key (rid) references route(rid)

);

create table payment (

payid int auto\_increment primary key,

ticketid int,

amount decimal(10,2),

pdate timestamp default current\_timestamp,

mode ENUM('online', 'counter') DEFAULT NULL,

refund enum('not applicable', 'pending', 'refunded') default 'not applicable',

foreign key (ticketid) references ticket(ticketid)

);

**TRIGGERS**

# Triggers in SQL are special procedures that automatically execute in response to certain events on a table, such as INSERT, UPDATE, or DELETE.

# 1. `after\_route\_insert`: Automatically inserts seat records into the `seat` table when a new route is added.

# 2. `after\_route\_delete`: Deletes seat records from the `seat` table when a route is removed.

# 3. `seatbook`: Updates the seat status to 'booked' when a ticket is inserted.

# 4. `refundset`: Sets the refund status to 'pending' when a ticket's booking status is updated to 'cancelled'.

# 5. `seatfree`: Updates the seat status to 'available' when a ticket's booking status is updated to ‘cancelled'.

PROCEDURES

1. `pnr`: Retrieves the booking status of a ticket using its PNR.

2. `seat\_details`: Fetches details of a seat using its seat ID.

3. `sched`: Provides the schedule of a train's journeys using its train ID.

4. `avail`: Lists available seats in a train by class.

5. `rac`: Lists RAC (Reservation Against Cancellation) seats in a train by class.

6. `all\_seats\_class`: Lists all non-booked seats in a train by class.

7. `all\_seats\_train`: Lists all non-booked seats in a train.

8. `passlist`: Retrieves details of all passengers on a specific train and date.

9. `waitlist`: Lists all waitlisted passengers on a train.

10. `refund`: Calculates the total refund amount for all canceled tickets on a train.

11. `revenue`: Calculates the total revenue for a train within a date range.

12. `cancelrec`: Lists PNRs and refunds for all canceled reservations.

13. `busyroute`: Identifies the most busy route based on ticket count.

14. `bill`: Retrieves the bill details (PNR, amount, mode) for a ticket using its PNR.

15. `Total\_due\_amount`: Calculates the total unpaid amount for a passenger.

16. `pay\_ticket`: Processes payment for a ticket by updating the payment mode.

17. `book\_ticket`: Books a ticket for a passenger by seat ID, updating the seat status and generating a PNR.

CALLS

# call pnr(1000000003);

call seat\_details(5);

call sched(2);

call avail(1,1);

call rac(1, 1);

call all\_seats\_class(1, 1);

call passlist(1, '2025-04-20');

call waitlist(1);

call refund(1);

call revenue('2025-04-01', '2025-04-30');

call cancelrec();

call busyroute();

call bill(1000000003);

call rac(1, 1);

call all\_seats\_class(1, 1);

call all\_seats\_train(1);

call Total\_due\_amount(1);

call pay\_ticket(1, 1, 'online');

call book\_ticket(1, 2);